

**Final Report**

**Grant No. NAG 9-188**

**12/1/87 to 11/30/88**

**MODELING AND SIMULATION OF A  
STEWART PLATFORM TYPE  
PARALLEL STRUCTURE ROBOT**

**by**

**Gee Kwang Lim  
Graduate Research Assistant**

**Robert A. Freeman  
Assistant Professor**

**Delbert Tesar  
Carol Cockrell Curran Chair in Engineering**

**THE UNIVERSITY OF TEXAS AT AUSTIN  
Mechanical Engineering Department  
Austin, Texas 78712  
512-471-3039**

**April 1989**

## Abstract

# MODELING AND SIMULATION OF A STEWART PLATFORM TYPE PARALLEL STRUCTURE ROBOT

by

Gee Kwang Lim, Graduate Research Assistant

Robert A. Freeman, Assistant Professor of Mechanical Engineering

Delbert Tesar, Carol Cockrell Curran Chair in Engineering

The kinematics and dynamics of a Stewart Platform type parallel structure robot(NASA's Dynamic Docking Test System) were modeled using the method of kinematic influence coefficients(KIC) and isomorphic transformations of system dependence from one set of generalized coordinates to another. By specifying the end-effector (platform) time trajectory, the required generalized input forces which would theoretically yield the desired motion were determined.

It was found that the relationship between the platform motion and the actuators motion was nonlinear. In addition, the contribution to the total generalized forces, required at the actuators, from the acceleration related terms were found to be more significant than the velocity related terms. Hence, the curve representing the total required actuator force generally resembled the curve for the acceleration related force. Another observation revealed that the acceleration related effective inertia matrix  $\begin{bmatrix} \dot{I}_{dd} \end{bmatrix}$  had the tendency to decouple, with the elements on the main diagonal of  $\begin{bmatrix} \dot{I}_{dd} \end{bmatrix}$  being larger than the off-diagonal elements, while the velocity related inertia power array  $\begin{bmatrix} \dot{P}_{dd} \end{bmatrix}$  did not show such tendency. This tendency results in the acceleration related force curve of a given actuator resembling the acceleration profile of that particular actuator. Furthermore, the investigation indicated that the effective inertia matrix for the legs is more decoupled than that for the platform. These observations provide essential information for further research to develop an effective control strategy for real-time control of the Dynamic Docking Test System.

PRECEDING PAGE BLANK NOT FILMED

## TABLE OF CONTENTS

	Page
ACKNOWLEDGMENTS -----	iv
ABSTRACT -----	v
TABLE OF CONTENTS -----	vii
 CHAPTER	
1 INTRODUCTION -----	1
2 BACKGROUND -----	5
2-1 Stewart Platform -----	5
2-1.1 Basic Design Concept -----	6
2-1.2 Control of the Mechanism -----	6
2-1.2.1 Linear Hydraulic Actuator -----	9
2-1.2.2 Articulated Levers -----	9
2-2 Dynamic Docking Test System(DDTS) -----	12
2-3 Survey of Related Work by Other Researchers -----	15
3 KINEMATIC AND DYNAMIC MODELING USING KINEMATIC INFLUENCE COEFFICIENTS -----	22
3-1 Overview of the Method of Kinematic Influence Coefficients -----	22
3-2 General Approach to Develop the Kinematic Model of Serial Manipulators -----	26
3-2.1 Kinematics of Serial Manipulators -----	32

	3-2.1.1 First-order Kinematics -----	34
	3-2.1.2 Second-order Kinematics -----	37
	3-2.1.3 Forward Kinematics -----	42
	3-2.1.4 Reverse Kinematics -----	43
3-3	General Approach to Develop the Dynamic Model of Serial Manipulators -----	46
	3-3.1 Applied Loads -----	47
	3-3.2 Inertial Loads -----	48
	3-3.3 The Dynamic Equations -----	53
4	TRANSFER OF GENERALIZED COORDINATES -----	55
4-1	Kinematic Model Transfer -----	55
	4-1.1 Direct Kinematic Model Transfer -----	56
	4-1.2 Indirect Kinematic Model Transfer -----	58
4-2	Dynamic Model Transfer -----	60
4-3	Application of Transfer of Generalized Coordinate to Multi-loop Parallel Mechanisms -----	63
5	KINEMATIC AND DYNAMIC MODELING OF THE DYNAMIC DOCKING TEST SYSTEM -----	72
5-1	Kinematic Model -----	72
	5-1.1 System Definition -----	74
	5-1.2 Specification of the Platform Position and Orientation -----	79
	5-1.3 Reverse Kinematics of Each Leg -----	81
	5-1.4 Initial First- and Second-order Kinematics of Each Leg -----	83
	5-1.5 First- and Second-order Kinematics of the Platform -----	84

5-2	Initial Dynamic Model of the Legs -----	93
5-3	Transfer of Generalized Coordinates to Obtain the Dynamic Model Referenced to the Platform -----	97
5-4	Transfer of the Dynamic Model to the Desired Set of Generalized Coordinates--The Input Actuators -----	101
6	APPLICATION OF THE MODEL -----	103
6-1	Verification of the Model -----	103
6-1.1	Special Case Model -----	103
6-1.2	Verification of the Result of the First-order KIC -----	107
6-1.3	Actuators Motion Verification -----	108
6-2	Simulation of the Dynamic Docking Test System(DDTS) -----	110
6-3	Conclusion -----	118
6-4	Summary -----	150

## APPENDIX

A	DEVELOPMENT AND DEFINITION OF GENERALIZED SCALAR (•) PRODUCT OPERATOR FUNDAMENTAL TO DYNAMIC MODELING AND TRANSFER OF COORDINATES -----	152
B	SIMULATION PROGRAM LISTING -----	155
C	INPUT DATA FOR THE SIMULATION PROGRAM IN APPENDIX B -----	200

D	SUBROUTINE FOR CLASS P=2, CONSTANT ACCELERATION -----	201
E	SUBROUTINE FOR CLASS P=4, 4-5-6-7 POLYNOMIAL -----	202
F	SUBROUTINE FOR CLASS P=4, THIRD DERIVATIVE TRAPEZOIDAL -----	203
G	SUBROUTINE TO DETERMINE THE CONSTANTS A1, A2 AND A3 -----	204
	REFERENCES -----	205

## CHAPTER 1

### INTRODUCTION

Change, rapid change is one of the most common daily occurrences in today's world. Everyday there are new products cropping up in the market. It is practically impossible for any manufacturer to stay competitive if the manufacturing facilities are solely designed for a particular product, requiring months or maybe years to convert the facilities for production of a different product. Hence, flexible manufacturing is the key word in today's manufacturing environment. The facilities must be able to adapt to changes rapidly. Besides being flexible in today's manufacturing world, certain manufacturing processes require an almost absolutely clean environment, eg., clean room operation. Furthermore, there are certain environments which are hazardous to human existence, such as operation in space and in nuclear facilities. In order to accomplish these tasks, robots, which are computer controlled mechanical devices capable of adapting to a wide range of operations, must be utilized. However, there is a limit to what a particular robot can or cannot do depending on the physical structure of the robot and how the robot is being controlled. There is nothing much that one can do to alter the physical structure of a robot, besides getting a different one. But, there are many ways in which one can improve the ability of a robot. The most obvious one is by altering the control strategy of the robot. For example, there are different ways for a robot end-effector to move to a certain location. One of the ways is to move the end-effector to the vicinity of the location as fast as possible and subsequently employ feedback control to reach the desired location. Another way is by using a more accurate model of the robot to predict the input loads required for a given motion of the end-effector and implement feedforward control in order to arrive at the desired location. The basis of this work then is to obtain a robot's dynamic models for the purpose of feedforward control.

There are a few desirable criteria which are frequently used to characterize robot manipulators. They are:

- (1) Load capacity;
- (2) Speed;
- (3) Precision.

These three important factors are very often interrelated. Robots with high load capacity are usually imprecise and can only move at low speed. Or, robots moving at high speed are often imprecise. Hence, there is a trade-off among the three criteria. There are three factors that characterize the precision of industrial robots. The first factor is repeatability. Repeatability refers to the ability of a robot to return to a previously defined location in space. The second factor is absolute accuracy, which refers to the ability of a robot to reach a point in space defined by the controller. The last factor is resolution. It gives the smallest movement the end-effector can achieve.

The main concern in this work is the problem of load capacity. The robots that are of interest here must be able to carry extremely high loads, maybe up to a few thousand kilograms. With this degree of load capacity, the robots themselves must be relatively rigid and heavy. As a result of the large mass in the robot structure, dynamic effects become very significant in the robot motions, although the robot maybe moving at a relatively low speed. There are basically two different classes of robots that are in use today. These are "serial" robot manipulators and "fully parallel" robot manipulators. Although, there can be a mix of the two classes, for example, the hybrid manipulator systems discussed in Sklar and Tesar [36], the focus in this work will be on serial and fully parallel manipulators.

Theoretically, serial robots can be designed to carry high load if necessary. Practically, it is almost impossible to design serial robots that can carry high loads and maintain relatively precise motion. This is due to the fact that errors in serial manipulators are additive in nature. A one degree deflection in the first link may easily cause a significant error at the end-effector position. On the other hand, parallel manipulators are structurally more rigid and the error in each links is non-additive. Therefore, parallel manipulators are the best alternative for tasks that require high load capacity in a limited workspace. However, parallel manipulators

not only are structurally more complex, but they also require a more complicated control scheme. The main objective of this work is to develop the necessary dynamic models and computer software for future study of the generalized Stewart Platform aimed at formulating an effective real-time control strategy.

This work concentrates on the development of the dynamic model for a fully parallel robotic manipulator based on the modeling technique called the method of Kinematic Influence Coefficients(KIC). The modeling procedure begins with the discussion on serial manipulators and then extends the technique to include parallel manipulators by utilizing an isomorphic transformation procedure called the transfer of generalized coordinates.

To provide the readers with a better idea of the discussion that focuses on the development of the model for a fully parallel manipulator, Chapter 2 begins with an introduction on how the concept of parallel mechanisms was first envisioned by Mr. D. Stewart [37] and the basic design concept for this class of mechanism, subsequently known as Stewart Platforms. Also included is the application employing this class of mechanism by the National Aeronautics and Space Administration(NASA) for the development of the Dynamic Docking Test System (Gates and Graves [18], Owen and Williams [33], Strassner ([38], [39], [40])). In addition, a brief discussion on the different approaches adopted by researchers to model Stewart Platform type mechanisms is meant to give the readers an overview of the problems faced when using this class of mechanism.

Chapter 3 introduces the fundamental modeling approach employed in this work. The approach, called the method of Kinematic Influence Coefficients, is based on the separation of time dependent functions and position(or configuration) dependent functions. This chapter develops the tools necessary for deriving the kinematic and dynamic models for serial manipulators. Then, using the tools just developed, the model for serial manipulators is formulated.

Chapter 4 gives a detailed description of the isomorphic transformation technique which greatly enhances the modeling capabilities of the method of Kinematic Influence Coefficients. A general procedure is developed to transfer the kinematic and dynamic models referenced to any set of generalized coordinates to any other desired set of generalized coordinates via the transfer of

generalized coordinates. The discussion begins with serial manipulators and then is extended to include multi-loop parallel mechanisms.

Chapter 5 focuses on the application of the modeling technique developed in the previous chapters to NASA's Dynamic Docking Test System, which is a variation of the generalized Stewart Platform. This chapter gives a detailed description of the derivation of the desired dynamic model, which can be used to implement feedforward control for the Dynamic Docking Test System. The development includes deriving the directly obtainable initial models for each leg referenced to their respective joint parameters and the platform model referenced to the platform coordinates. Also included is the transfer of the joint-based model for each leg to the intermediate common set of platform coordinates. And, finally, the transfer of the model from the platform coordinate set to the desired input coordinate set is discussed.

In Chapter 6, the results of various computer simulations using the model developed in Chapter 5 are presented. Four different motion specifications and two platform trajectories are simulated. The motion specifications used are:

- (a) Class  $p=2$ , constant acceleration;
- (b) Class  $p=3$ , 3-4-5 polynomial;
- (c) Class  $p=4$ , 4-5-6-7 polynomial;
- (d) Class  $p=4$ , 3<sup>rd</sup> derivative trapezoidal.

From these simulations, the contribution to the overall actuator forces arising from the velocity-related term and the acceleration-related term are studied. Furthermore, the elements in the effective inertia matrix and inertia power array are also investigated. The magnitude of these elements in general and the relative magnitude of the main diagonal elements compared to the off-diagonal elements, are essential for studying the feasibility of real-time feedforward control.

## CHAPTER 2

### BACKGROUND

#### 2.1 Stewart Platform

In the search for a suitable means for simulating flight conditions for the safe training of helicopter pilots, the design of a mechanism has been established having all the freedoms of motion within the design limitations of amplitude and capable of being controlled in all of them simultaneously.

In the opening paragraph, as quoted above, from the 1965 publication by Mr. D. Stewart [37], he saw the immediate need for a mechanism suitable for pilot training. Serial manipulators having six degrees-of-freedom(DOF) are potential candidates for the job. However, the mechanism must be able to carry a large load and change direction, speed and acceleration within a short period of time to simulate flight conditions. These requirements present a serious problem for any serial manipulator. It was a known fact at that time, and today too, that serial manipulators were very limited in load capacity. Thus, using a serial manipulator alone to do the job is highly unlikely, so a different kind of mechanical structure is necessary. Mr. Stewart envisioned mechanisms which had the ability to carry a large load and still satisfy the above mentioned motion specifications could also be used in the following ways:

- (a) To simulate a space vehicle
- (b) To simulate a stationary platform on a moving ship
- (c) As a human control mechanism (man-machine interface)
- (d) As a machine tool
- (e) As an automatic assembly or transfer machine.

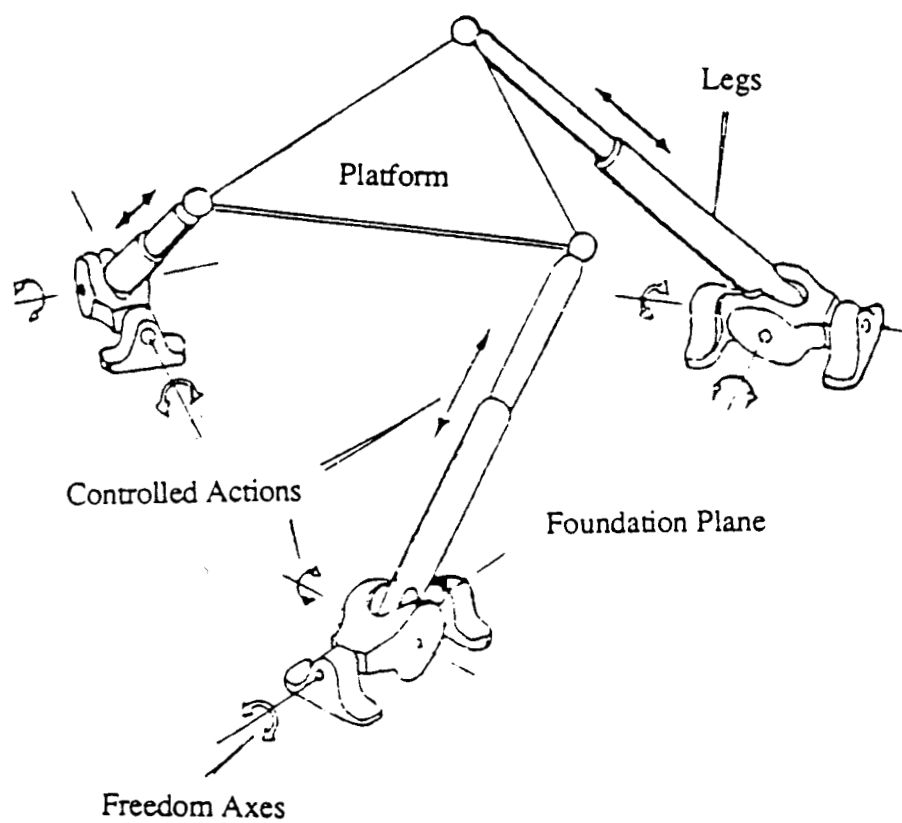
### 2.1.1 Basic Design Concept

The mechanism which Mr. Stewart designed, called the Stewart Platform, consists of a moving platform supported by three legs through a ball joint at each of the connections. The other end of each leg is connected to ground through a two-axis revolute joint as shown in Fig. 2-1. The connections at the ball joints are free to rotate as the platform moves. All the legs are designed using prismatic joints allowing control of the individual leg lengths. One axis of the two-axis revolute joint is also controlled by an actuator. These inputs are referred to as the controlled actions in Fig. 2-1. This arrangement gives the platform six DOF with two DOF essentially controlled by each leg. The platform is also designed so that when the three legs are in their mean positions, each of the legs is contained in a tangential plane of a circle that passes through the three points of the platform as shown in Fig. 2-2.

With this basic design concept, a wide range of applications previously limited by the speed and load capacity of serial manipulators will soon surface. For example, applications that require high load capacities will now be possible due to the parallel nature of the mechanism. Unlike serial manipulators where deflection from each link is additive in nature, any deflection under heavy load in a parallel manipulator is non-additive, meaning the total deflection at the end-effector is not the sum of each of the individual link. Another useful application is in the area where high precision and accuracy are critical within a limited operating range, eg. micromanipulator for use in micro-surgery. In this kind of operation, very small motions are required from the manipulator but with high precision, the parallel structure potentially gives the mechanism a faster and more precise motion than its serial counterpart.

### 2-1.2 Control of the Mechanism

As technology advances, system control becomes a vital part of everyday life. Take the very basic household temperature control. Without the use of a thermostat, one will have to perform the boring routine of switching on and off



**Fig. 2-1 General Arrangement of the Stewart Platform\***

\*Adapted from Stewart [37]

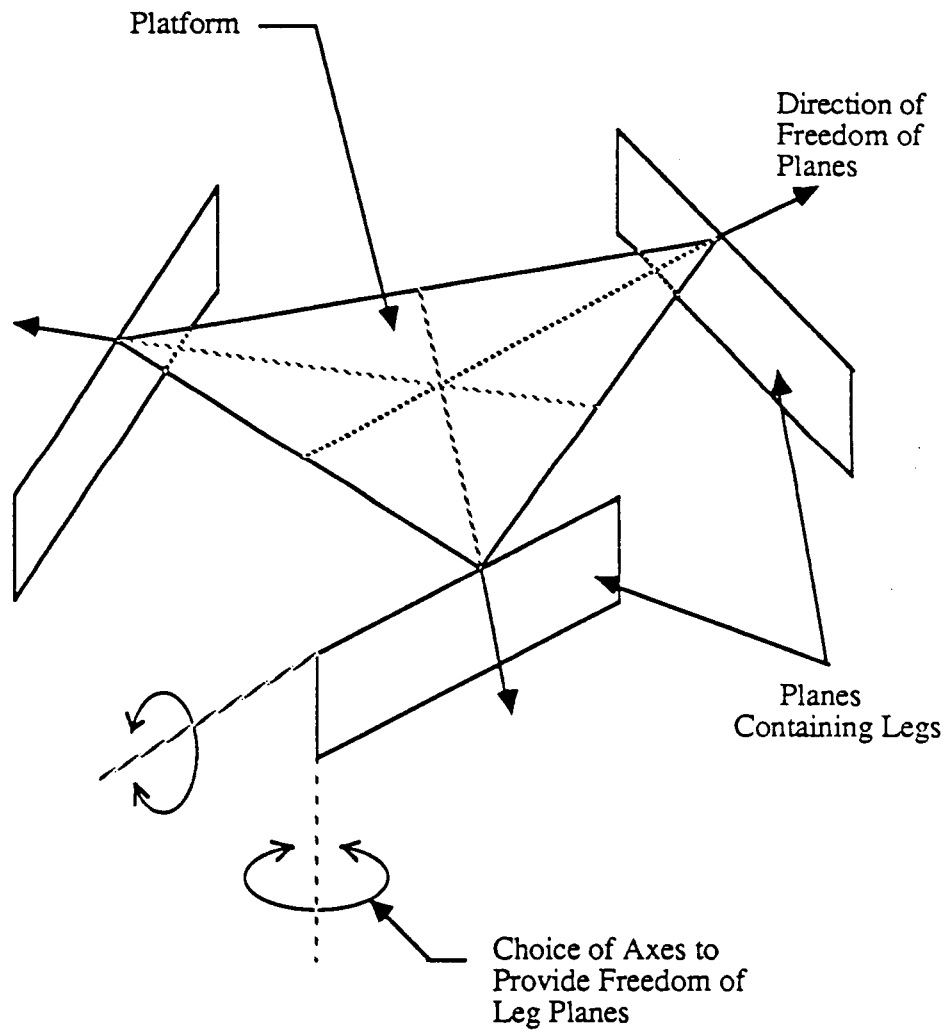


Fig. 2-2 Diagram of Leg Planes

the air conditioner(or heater) to keep the temperature at a desired level. Opening and closing of the elevator door is another example of control. When someone enters the elevator as the door is closing, the door will stop and start to open again. These examples show how important control is in one's daily life, although one might not realize it. Control systems not only relieve humans from some easy tasks, but they can also facilitate the operation of a highly complex problem like sending a spacecraft out to an unknown galaxy. For any mechanism be of any use, the ability to control it effectively is essential. A mechanism will just be a show piece with no practical purpose if there is no way of controlling it. Hence, this section discusses some of the techniques suggested by Mr. Stewart for the control of the platform mechanism.

#### 2-1.2.1 Linear Hydraulic Actuator

One of the methods that Mr. Stewart suggested in controlling the mechanism was by using two hydraulic jacks for each leg. One of the jacks is to control the length of the leg, while the other jack is to control the angle of the first jack as shown in Fig. 2-3. Also depicted in the figure is a common axis and two parallel axes at the base of the two jacks (two-axis joint). The common axis is not controlled by the leg allowing the plane containing the individual leg to rotate freely about that axis. This design gives the platform a three-axis motion about the ball joint. When the three legs are connected together at the platform, the platform direction, which cannot be controlled by a single leg, can be thought of as being controlled by the other two 2 DOF legs. Thus, the platform has a total of six DOF (three translational and three rotational motions).

#### 2-1.2.2 Articulated Levers

Another proposed structure uses articulated levers as shown in Fig. 2-4. This system of controlling the platform differs from the linear actuators discussed in the previous section. Instead of controlling one length and an angle, the ball joint location within the plane of each leg, and hence, the platform is controlled by two angles ( $\alpha$  and  $\beta$ ). One advantage for controlling two angles over

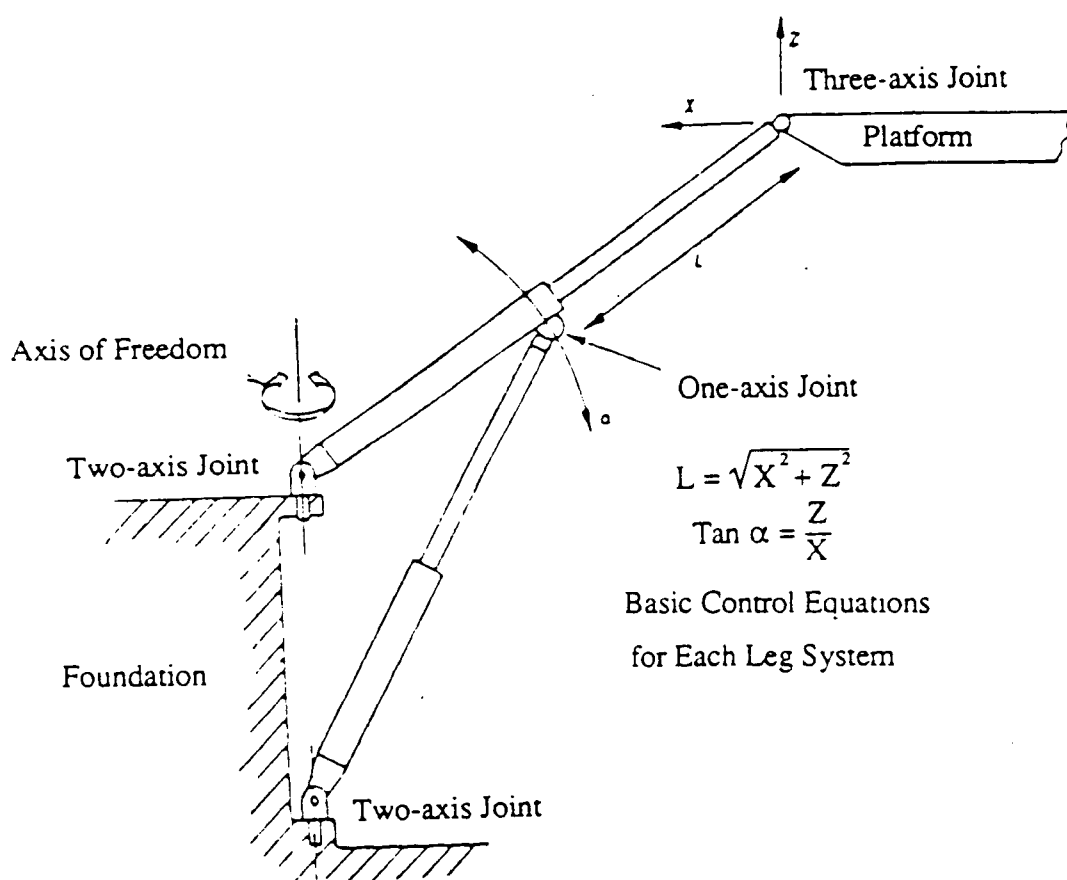


Fig. 2-3 General Arrangement of Single Leg System\*

\*Adapted from Stewart [37]

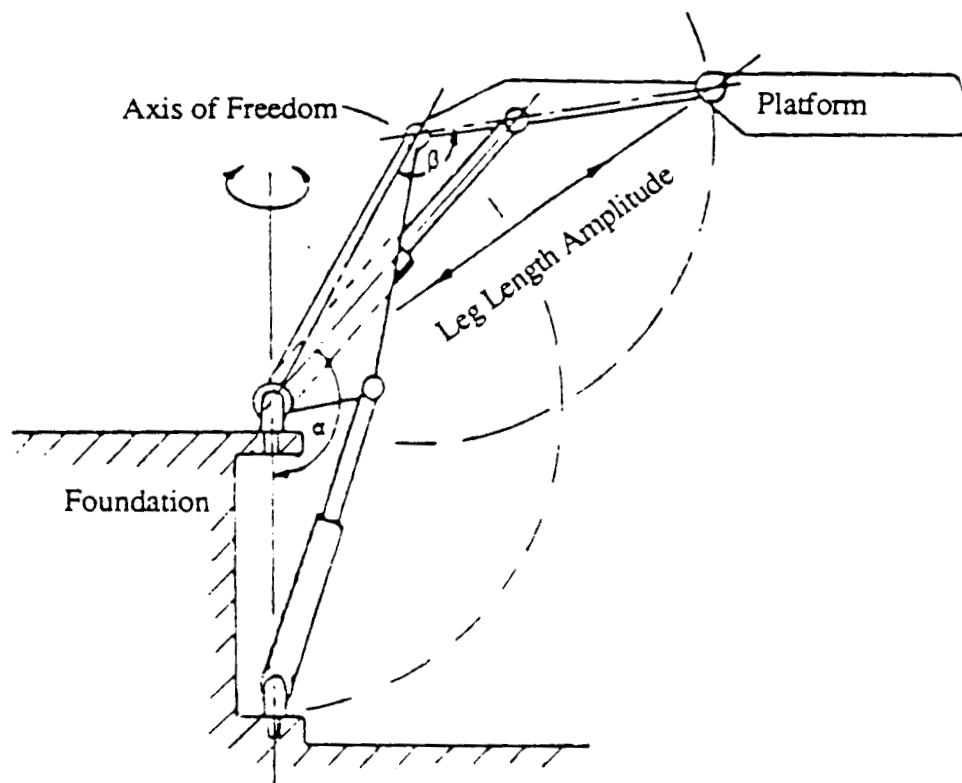


Fig. 2-4 Articulated Levers\*

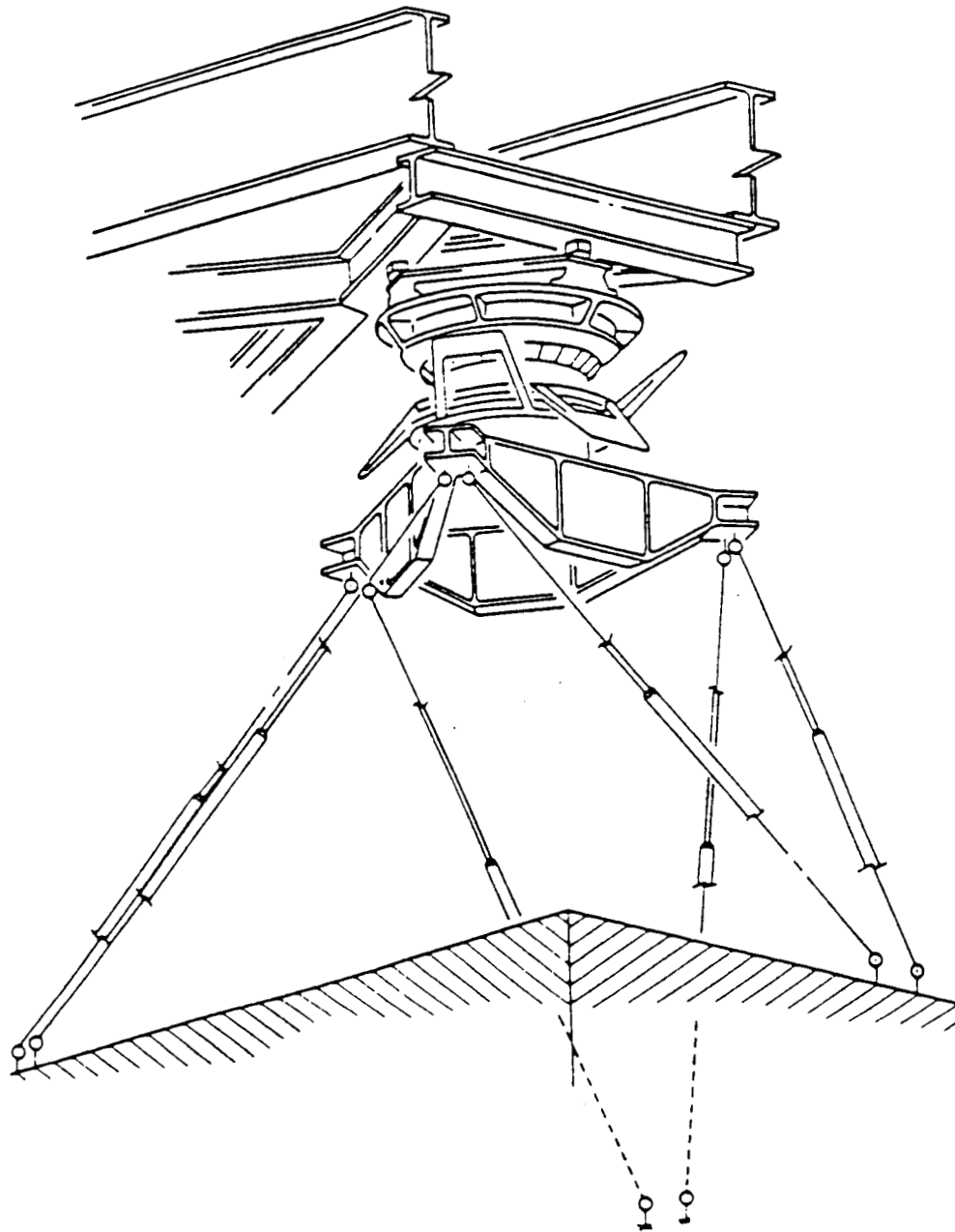
\*Adapted from Stewart [37]

one length and an angle, as discussed in section 2-1.2.1, is that larger workspace can be attained. By replacing the jack in Fig. 2-3 with an articulated leg, and operating the jack part way along the outer limb (Fig. 2-4), the amplitude of the leg extension can be increased as compared with the linear controlled actuator.

## 2.2 Dynamic Docking Test System (DDTS)

Utilizing the basic concept of the Stewart Platform, the National Aeronautics and Space Administration's Lyndon B. Johnson Space Center built a full-scale advanced docking system in the early 1970's (Gates and Graves [18], Owen and Williams [33], Strassner ([38], [39], [40])). The DDTS is a large motion, real-time docking test simulator designed to physically accommodate the docking hardware of two spacecrafts. The physical configuration of the simulator is shown in Fig. 2-5. The simulator consists of six linear hydraulic actuators which support and move an active table (moving platform) on which the passive docking hardware is mounted during a simulation. The overhead support structure is supported by a strongback at one end and two vertical braced columns at the other as shown in Fig. 2-6. This stationary portion of the simulator supports the active docking hardware system, docking hardware adapter and load cell system.

During a docking test, the lower portion of the simulator manipulates the passive docking hardware of one spacecraft to the active docking hardware of the other spacecraft, which is attached to the stationary overhead structure. Before the two portions of the simulator (ie., active and passive) come into contact, the active table is driven by the six hydraulic actuators in a preprogramed motion trajectory relative to the stationary active docking hardware. On contact, the loads sensed by the load cell system triggers the closed-loop portion of the simulation. These loads are then used as inputs to spacecraft equations of motion to predict the response of the two spacecrafts on the computer. Subsequently, real-time relative motion of the two spacecrafts is determined and then transformed into actuator motion commands. The simulation is terminated when the closed-loop load and the spacecraft dynamics go to zero.



**Fig. 2-5 Perspective View of the Dynamic Docking Test System\***

\*Adapted from Strassner [38]

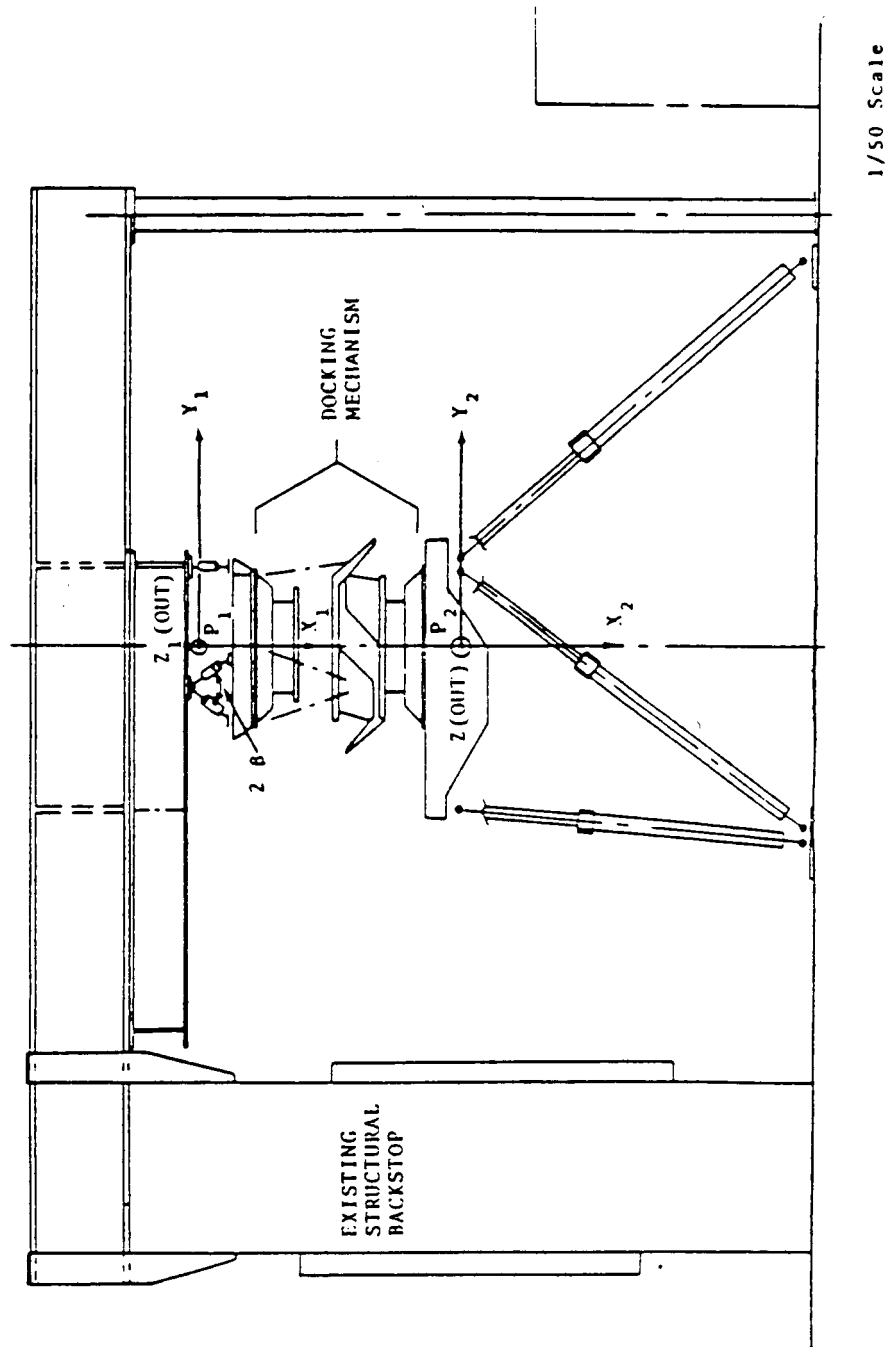


Fig. 2-6 The Dynamic Docking Test System\*

\*Adapted from Strassner [38]

Ignoring the specific structural differences between the Stewart Platform of Fig. 2-1 and the DDTS of Fig. 2-5, it is apparent that the two mechanisms resemble each other. Both the mechanisms have a moving platform that is supported by some mechanical linkages (or legs) and they are parallel in nature. By controlling the orientation and/or length of the legs, the motion of the platform is then defined. The only difference between the two mechanisms is that the platform in Fig. 2-1 has three legs while the platform in Fig. 2-5 has six legs. However, both the mechanisms still provide control of the six DOF of the platform. Two DOF are controlled by each of the leg in the arrangement in Fig. 2-1, but only one DOF is controlled by each of the leg in the design shown in Fig. 2-5.

The initial use of the DDTS was for the Apollo/Soyuz Test Project, which was an international mission between the United States and the USSR. The project involved the docking of Apollo and Soyuz docking modules in the space orbit. The high load capacity requirement from the two docking modules discouraged the use of serial test mechanism due to the additive nature of the deflection from each of the serial linkages. Future applications of the DDTS will be for the space shuttle and the space station.

### 2.3 Survey of Related Work by Other Researchers

In the last two decades, science and technology have advanced tremendously. Computer technology for instance, has reached a point where products produced just one or two years ago are considered outdated. With this in mind, one would expect the same impact on robotic technology since the computer is the brain of the robot. Unfortunately, this is not true. Only recently has industry seen the potential for robots in the manufacturing environment. This delay in realizing the need for robotics technology created a gap between what the brain of a robot is capable of doing and what the robot is physically able to do. Although there is a lot of robotics research ongoing, most of the emphasis is in the area of serial manipulators, from which most industrial robots are derived. A very limited amount of effort is put into the development of parallel robotic mechanisms. A survey of the literature will illustrate this trend in the engineering research community. After an

extensive survey, only the following publications are available: Callan [4], Cox [6], Cwiakala [8], Do and Yang [9], Fichter [12], Freeman and Tesar ([14], [15]), Hudgens [21], Hunt [22], Marco [28], Mayer and Wood [30], Mohamed and Duffy [31]. This limited effort maybe due to the complexity of the geometry as compared to serial linkages or, it maybe because not many researchers have yet realized the potential of parallel mechanisms.

In the following, a few of the recent publications from researchers across the country covering the different aspects of parallel mechanisms will be discussed. The first report investigates the dynamics of the platform type mechanism using the Newton-Euler formulation. Following that is a report that investigates the workspace of the mechanism. The third report deals with the general theory and practical construction of the mechanism. While the last investigation is to utilize the mechanism as a micromanipulator for delicate force control, error compensation and fine manipulation.

In a publication presented by Do and Yang [9], Newtonian mechanics was used to calculate the actuating forces for the actuators of a Stewart Platform shown in Fig. 2-7. It is found that the dynamics of the mechanism is governed by thirty-six simultaneous equations. Instead of solving the entire set of simultaneous equations, which would be computationally demanding, it is also found that the thirty-six equations can be arranged to form systems of six linear equations. The approach taken in the reported research is very "conventional". By conventional it is meant that the method is widely used in the engineering community. As a result, it is easily understood by most researchers. However, this approach lacks generality in the sense that there is no general rule as to how to form the six desired equations. A complete analysis of the mechanism may be necessary to obtain a similar set of six equations if a slightly different model is used.

Another publication presented by Cwiakala [8], analyzed the kinematics of the platform. Using only the kinematic model, the workspace of the Stewart Platform was studied. By utilizing the special symmetry of the platform, the investigator found that the problem of determining a representative workspace cross-section could be reduced to a planar problem. Also addressed in this work is the development of an efficient approach to generate the workspace of the platform.

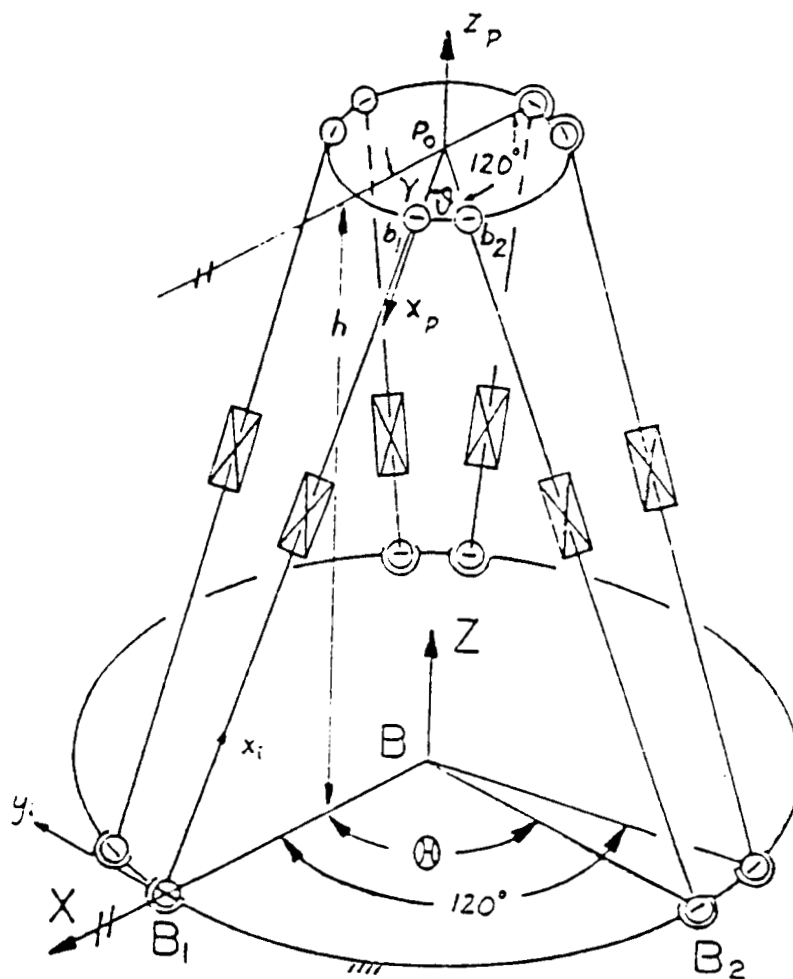


Fig. 2-7 Kinematic Model of the Stewart Platform\*

\*Adapted from Do and Yang [9]

It is found that the workspace cross-section of the Stewart Platform under consideration is nonsymmetric. In addition, the nonsymmetry and the volume increase as the ratio of the lower radius (base) to upper radius (platform) increases. This implies that by increasing the radius of the platform, the workspace can be increased.

A different approach was taken by Fichter [12]. Screw theory was used to determine the dynamics of the Stewart Platform. Unfortunately, the author of this work is not familiar enough with the technique of screw theory to contribute any useful comment. However, one of the findings which has great interest to this author is the determination of singular positions for the mechanism. It is noted that instead of losing one or more degrees of freedom at the singular positions, as one does with serial manipulators, parallel manipulators gain one or more degrees of freedom. By this it means that the control of one or more degrees of freedom of the platform is lost. The singular positions of the Stewart Platform similar to the one shown in Fig. 2-5 or Fig. 2-7 occur when the six lines of action of the forces of the legs are linearly dependent. This condition can also be found by calculating the determinant of the matrix of the Plucker coordinates. At singular positions, the determinant of the matrix becomes zero. For detailed results of this investigation the readers are referred to the publication. In addition, Mohamed and Duffy [31] also investigated the first-order properties via screw theory.

Applying the concept of the Stewart Platform, researchers have come up with many applications for parallel robotic mechanisms. One of the very practical uses is as a micromanipulator. It can be used alone for very small motion and high precision operation or it can be combined with a serial manipulator to take advantage of the positive aspects of both the parallel and serial manipulators. By positive aspects the author means the relatively high speed and precision of the parallel structure, and large workspace volume together with the relatively high dexterity of the serial structure. Sklar and Tesar [36] investigated the kinematics and dynamics of hybrid serial/parallel manipulator structures.

Hudgens [21] investigated a fully-parallel six DOF micromanipulator in his Master's Thesis. The micromanipulator investigated there, shown in Fig. 2-8, is a variation of the generalized Stewart Platform. Instead of varying the link

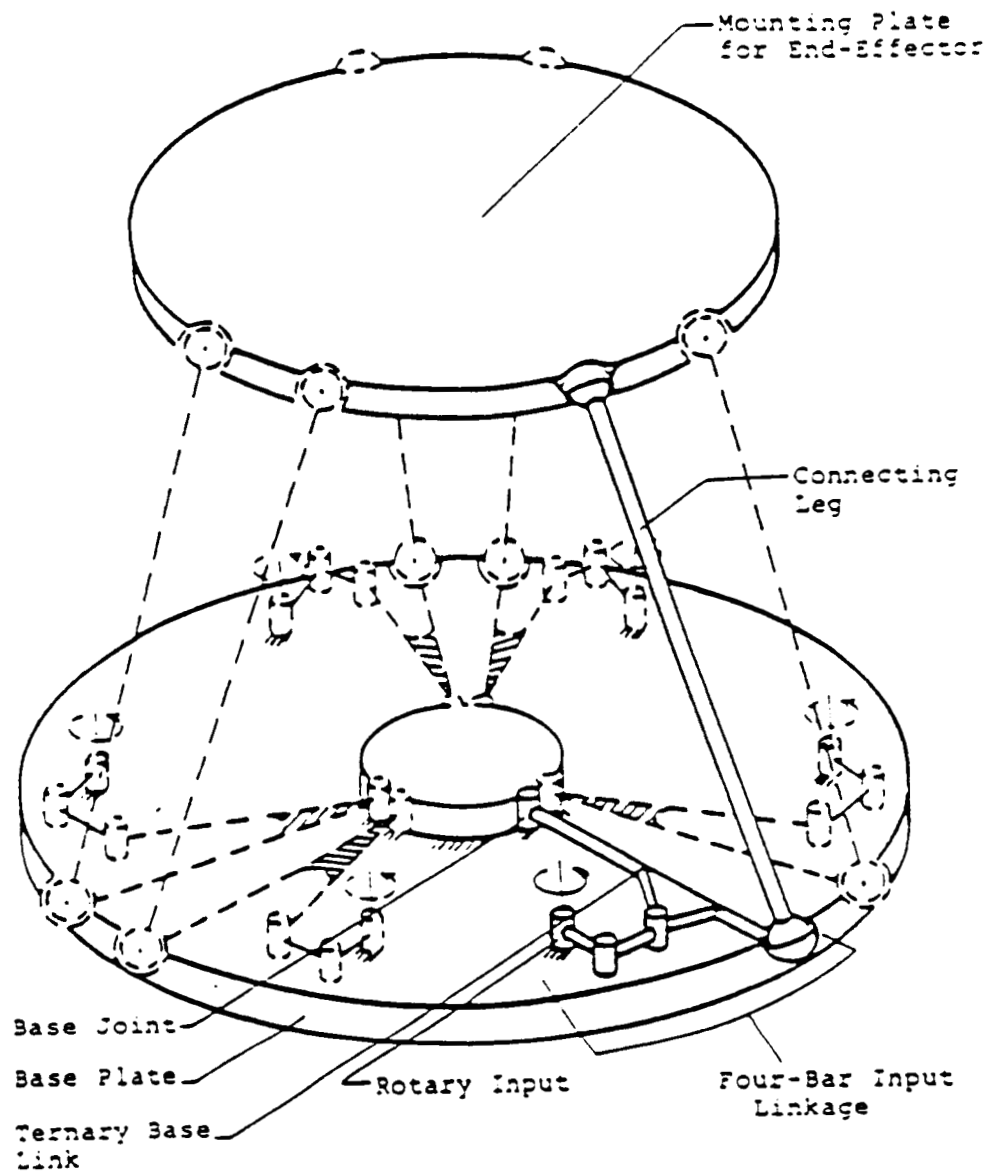


Fig. 2-8 Kinematic Representation of the  
Micromanipulator Mechanism\*

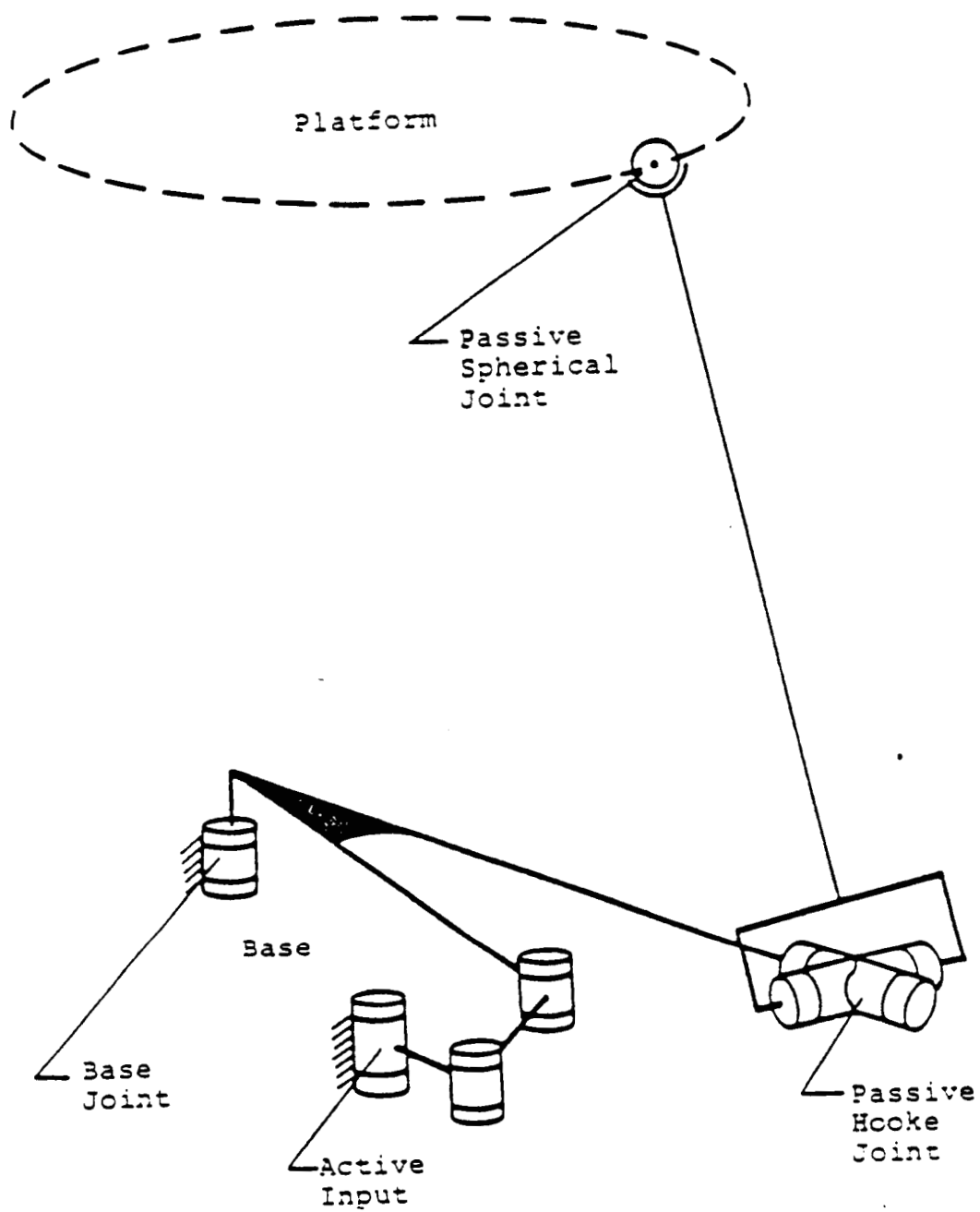
\*Adapted from Hudgens [21]

length of the six legs to control the platform, it is controlled by varying the position of the lower spherical joints in a circular arc within the base plane. The rotary input in Fig. 2-8 controls the movement of the lower spherical joint via the base link. The motivations for using the four-bar linkage at the active input to control the platform instead of the link length are the following:

- (a) only a small workspace is necessary in this particular application.
- (b) this input arrangement provides high resolution at the platform. By high resolution it means large input motion to small platform motion.
- (c) the fixed link provides higher load capacity compared to prismatic link.

One simplification made in his research is to replace the lower spherical joints by two DOF hooke (universal) joints, since the spin of each leg about its own axis does not contribute to or affect the input/output relationship of the mechanism. This simplification is shown in Fig. 2-9 for one branch. The modeling technique employed by Mr. Hudgens is very similar to the method used in this work. However, numerous modifications and variations are made and will be highlighted as they appear in the following chapters of this work.

This chapter is only intended to provide a quick and brief overview of the past and present works in the area of parallel robotic mechanisms. It in no way represents all the research and advances in this area. However, deriving from the limited resources and knowledge, the author strongly feels that a lot of research is needed in order to fully understand and exploit the potential of parallel mechanisms.



**Fig. 2-9 Kinematic Representation of a Single Branch\***

\*Adapted from Hudgens [21]

## CHAPTER 3

### KINEMATIC AND DYNAMIC MODELING USING KINEMATIC INFLUENCE COEFFICIENTS

#### 3-1 Overview of The Method of Kinematic Influence Coefficients

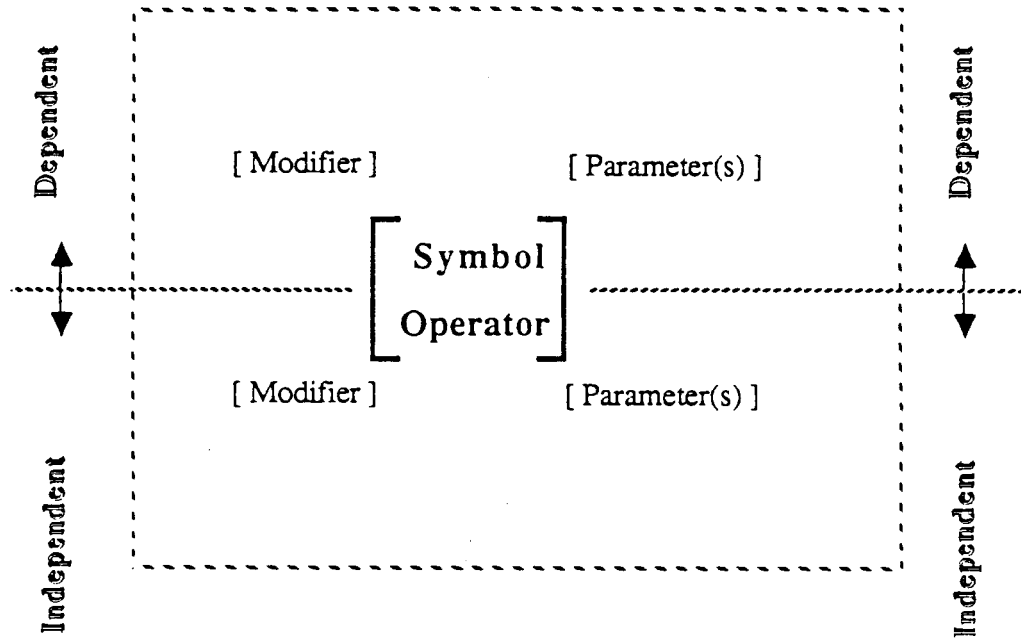
Section 2-3 reviewed some of the techniques employed by a few researchers for the kinematic and dynamic modeling of six DOF parallel robotic mechanisms. This chapter will discuss a technique not very widely used in the engineering community but which has been used with great success by a group of researchers at the University of Florida(now at The University of Texas at Austin) for the past few decades. The technique has been termed the method of Kinematic Influence Coefficients(KIC). The basis of this work stems from the method of KIC continually developed over the years by researchers including Tesar, Benedict, Thomas and Freeman. The present chapter will concentrate on the discussion of serial manipulator modeling using KIC. In the next chapter, discussion will focus on how to apply the model of a serial manipulator to a parallel manipulator. This ability is largely a result of the generality and versatility of the KIC approach.

The method of KIC is based on the separation of time dependent functions and position(or configuration) dependent functions. Throughout the entire process of obtaining the kinematic and dynamic models, the formulation will strictly adhere to this fundamental concept, keeping the two terms separated. Following this concept, a lot of research has been done and well documented in the literature. Some of the more significant findings are : Benedict and Tesar ([1], [2], [3]), Thomas and Tesar ([42], [43]), Freeman and Tesar ([13], [14], [15]). The technique was initially developed by Benedict and Tesar to analyze planar mechanism. In 1982 Thomas and Tesar took the approach one step further by developing procedures for the analysis of a general serial manipulator. Freeman and Tesar through the years 1982 to 1986 extended the applicability of the approach by developing the technique of generalized coordinate transformation, which will be

discussed in the next chapter. This greatly enhances the potential power of KIC in dealing with a larger and more general class of mechanism and forms the analytic basis of this work.

To aid readers who are unfamiliar with the method of KIC and the notational scheme employed in this work, it is essential to introduce the readers to the notation used in the present work. The notational scheme adapted here was developed by Freeman and Tesar [14]. Although part of the notation may appear redundant at the stage when dealing with a single set of independent generalized coordinates, it will be shown to graphically enhance the manipulation process for the transfer of generalized coordinates, which will be discussed in the following chapter. The basic formulation of this scheme involves a square(or block) arrangement with the central block surrounded by both pre- and post- superscripts and subscripts at the four corners as illustrated in Table 3-1. The square is divided into two portions with the top half reserved for dependent system parameters and the bottom half reserved for independent system parameters. The center of the square is reserved for a symbol representing a system parameter (e.g., a set of generalized position parameters ( $\underline{u}$ )), a physical quantity (e.g., a generalized force ( $\underline{T}$ )), or a mathematical operation (e.g., first-order partial geometric derivative ( $G$ )). Post super- and sub-scripts denote which system parameter is involved with the center symbol. Pre super- and sub-scripts give any additional information that might be helpful to describe the system parameter. Hence, at times the pre scripts maybe missing from the square. Matrices and higher dimensional arrays are denoted by square brackets(ie.  $[ ]$ ) enclosing the symbol along with the super- and sub-scripts. Although the notational scheme may appear complex and confusing to first time readers, when one becomes familiar with its usage, the advantages of this scheme will soon surface. One will greatly appreciate the graphically descriptive information provided in the square when dealing with the transference of system dependence from one set of generalized coordinates to another.

Also, additional notation is used when dealing with higher dimensional arrays and their subsets. When four dimensional arrays are used in this work, the first index represents the leg number associated. Unfortunately, the first index of a three dimensional arrays may represent the leg number or plane



### Examples

1.  $\underline{u} = f(\varphi)$  :

$$\text{Dependent : } \underline{u} = \left( u^1, u^2, \dots, u^p \right)^T \quad \left\{ \begin{array}{ll} [ ] & [ p ] \\ [ u ] & [ ] \\ [ ] & [ ] \end{array} \right.$$

$$\text{Independent : } \underline{\varphi} = \left( \varphi_1, \varphi_2, \dots, \varphi_M \right)^T \quad \left\{ \begin{array}{ll} [ ] & [ ] \\ [ \varphi ] & [ ] \\ [ ] & [ m ] \end{array} \right.$$

Table 3-1 Notational Scheme

$$2. \frac{\partial(u)}{\partial \varphi} = \frac{\begin{bmatrix} \phantom{u} \end{bmatrix}}{\begin{bmatrix} \phantom{\varphi} \end{bmatrix}} \frac{\begin{bmatrix} \phantom{u} \end{bmatrix}}{\begin{bmatrix} \phantom{\varphi} \end{bmatrix}} \equiv [G_{\varphi}^u]$$

$$\text{where } G \equiv \frac{\partial(\phantom{u})}{\partial(\phantom{\varphi})}$$

$$3. \frac{\partial^2(u)}{\partial \varphi^2} = \frac{\begin{bmatrix} \phantom{u} \end{bmatrix}}{\begin{bmatrix} \phantom{\varphi} \end{bmatrix}} \frac{\begin{bmatrix} \phantom{u} \end{bmatrix}}{\begin{bmatrix} \phantom{\varphi} \end{bmatrix}} \frac{\partial^2(\phantom{u})}{\partial(\phantom{\varphi}) \partial(\phantom{\varphi})} \equiv [H_{\varphi\varphi}^u]$$

$$\text{where } H \equiv \frac{\partial^2(\phantom{u})}{\partial(\phantom{\varphi}) \partial(\phantom{\varphi})}$$

$$4. \sum_{j=1}^M \left\{ M_{jk} [{}^j G_{\varphi}^c]^T [{}^j G_{\varphi}^c] \right\} = \frac{\begin{bmatrix} \phantom{u} \end{bmatrix}}{\begin{bmatrix} \phantom{\varphi} \end{bmatrix}} \frac{\begin{bmatrix} \phantom{u} \end{bmatrix}}{\begin{bmatrix} \phantom{\varphi} \end{bmatrix}} [I^{\bullet}] \equiv [I_{\varphi\varphi}^{\bullet}]$$

$$\text{where } I_{(\varphi)\varphi}^{\bullet} \equiv \sum_{j=1}^M M_{jk} [{}^j G_{(\varphi)}^c]^T [{}^j G_{(\varphi)}^c]$$

Table 3-1 Notational Scheme(cont.)

associated. However, only in the simulation software are four dimensional arrays used with the first index indicating the leg number. Otherwise, throughout the derivation, only up to three dimensional arrays are used with the first index indicating the plane number. The last two indices represent the row and column, respectively, of a matrix or the matrix of one of the planes. In other words, the indices for four dimensional arrays, from left to right denote: leg; plane; row; column. For three dimensional arrays, the indices denote: leg/plane; row; column. Table 3-2 gives an example of the notation used. The subset of an array is also indicated by its indices. A missing index implies that all the components in that dimension are present in the subset of the array. For example, an array with indices like this:  $k; ;m;n$ , represents the  $m^{\text{th}}$  row and  $n^{\text{th}}$  column of all the planes for the  $k^{\text{th}}$  leg. This is also illustrated in Table 3-2.

The next two sections will focus on the development of a general approach for the kinematic and dynamic modeling of serial manipulators, based mainly on Freeman and Tesar [14] and Hudgens and Tesar [21]. Due to the fact that all parallel mechanisms can be viewed and modeled as a combination of serial mechanisms, it is essential that the readers fully understand the derivation that will be discussed in the following sections. Besides, almost all of today's industrial robots are serial in nature, hence, the method discussed here can be used to analyze almost any robot in use and to aid in the design of robots.

### 3-2 General Approach to Develop the Kinematic Model of Serial Manipulators

Consider a set of M-dimensional time dependent motion parameters, eg., the vector ( $\underline{u}$ ), written as

$$\underline{u}(t) = \left[ u^1(t), u^2(t), u^3(t), \dots, u^M(t) \right]^T, \quad (3-1)$$

where the superscripts(1,2,3,...,M) denote which of the system parameters are involved in the description of the kinematic state. The superscript T denotes the

$$[A]_{::} = \begin{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} & \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \\ \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} & \begin{bmatrix} d_{11} & d_{12} \\ d_{21} & d_{22} \end{bmatrix} \end{bmatrix}$$

Link 1                      Link 2

$$[A]_{1::} = \begin{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \\ \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} \end{bmatrix}$$

$$[A]_{1;2::} = \begin{bmatrix} c_{11} & c_{12} \\ c_{21} & c_{22} \end{bmatrix} ; [A]_{1;2;1} = \begin{bmatrix} c_{11} \\ c_{21} \end{bmatrix} ; [A]_{1;2;1;2} = c_{21}$$

Example of the A array with indices like this : 2 ; ; 1;1, represent

$$[A]_{2;;1} = \begin{bmatrix} b_{11} \\ d_{11} \end{bmatrix}$$

For three dimensional array A,

$$[A]_{;;} = \begin{bmatrix} \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix} \\ \begin{bmatrix} b_{11} & b_{12} \\ b_{21} & b_{22} \end{bmatrix} \end{bmatrix}$$

$$[A]_{1;;} = \begin{bmatrix} a_{11} & a_{12} \\ a_{21} & a_{22} \end{bmatrix}$$

**Table 3-2** Indicical Notation for Higher Dimensional Array

transpose of the row matrix. It is also assumed that  $\underline{u}$  is a function of an N-dimensional set of generalized coordinates  $\underline{\varphi}$ ,

$$\underline{\varphi}(t) = [\varphi_1(t), \varphi_2(t), \varphi_3(t), \dots, \varphi_N(t)]^T, \quad (3-2)$$

where the subscripts denote which of the generalized coordinates are involved. As a result, the system can be expressed parametrically as

$$u^m = u^m(\underline{\varphi}) ; m = 1, 2, 3, \dots, M \quad (3-3)$$

and

$$\varphi_n = \varphi_n(t) ; n = 1, 2, 3, \dots, N. \quad (3-4)$$

Using the above notation, the first order time derivative of  $\underline{u}$  is

$$\begin{aligned} \frac{d\underline{u}}{dt} &= \frac{\partial \underline{u}}{\partial \underline{\varphi}} \frac{\partial \underline{\varphi}}{\partial t} \\ &= \frac{\partial \underline{u}}{\partial \underline{\varphi}} \dot{\underline{\varphi}} \end{aligned} \quad (3-5)$$

By defining the first order kinematic influence coefficients, which will be referred to as the "G-function" throughout this work (as opposed to the more common use of the term Jacobian), as

$$[G_{\varphi}^u] = \frac{\partial \underline{u}}{\partial \underline{\varphi}} \quad (3-6)$$

equation 3-5 can be expressed as

$$\dot{\underline{u}} = [G_{\varphi}^u] \dot{\underline{\varphi}} \quad (3-7)$$

where  $[G_{\varphi}^u]$  is a M by N matrix and  $\dot{\underline{\varphi}}$  is a N by 1 vector.

The second order time derivative of  $\underline{u}$  is a direct differentiation of  $\dot{\underline{u}}$ , which is expressed as

$$\ddot{\underline{u}} = \frac{\partial \underline{u}}{\partial \underline{\Phi}} \ddot{\underline{\Phi}} + \frac{d}{dt} \left( \frac{\partial \underline{u}}{\partial \underline{\Phi}} \right) \dot{\underline{\Phi}}$$

or

$$\ddot{\underline{u}} = \frac{\partial \underline{u}}{\partial \underline{\Phi}} \ddot{\underline{\Phi}} + \frac{d}{dt} \left( \left[ G_{\Phi}^u \right] \right) \dot{\underline{\Phi}} \quad (3-8)$$

Applying the chain rule to the time derivative of  $\left[ G_{\Phi}^u \right]$  gives

$$\begin{aligned} \frac{d}{dt} \left[ G_{\Phi}^u \right] &= \frac{d}{dt} \left( \frac{\partial \underline{u}}{\partial \underline{\Phi}} \right) \\ &= \frac{\partial}{\partial \underline{\Phi}} \left( \frac{\partial \underline{u}}{\partial t} \right) \end{aligned} \quad (3-9)$$

Substituting equation 3-5 into equation 3-9 and regrouping gives

$$\begin{aligned} \frac{d}{dt} \left[ G_{\Phi}^u \right] &= \left( \frac{\partial}{\partial \underline{\Phi}} \frac{\partial \underline{u}}{\partial \underline{\Phi}} \right) \dot{\underline{\Phi}} \\ &= \left( \frac{\partial}{\partial \underline{\Phi}} \left[ G_{\Phi}^u \right] \right) \dot{\underline{\Phi}} \end{aligned} \quad (3-10)$$

By definition, the second-order kinematic influence coefficient is the partial derivative of  $\left[ G_{\Phi}^u \right]$  with respect to  $\underline{\Phi}$ , which will be denoted throughout this work as

$$\left[ H_{\Phi\Phi}^u \right] = \left( \frac{\partial}{\partial \underline{\Phi}} \left[ G_{\Phi}^u \right] \right) = \left( \frac{\partial}{\partial \underline{\Phi}} \frac{\partial \underline{u}}{\partial \underline{\Phi}} \right) \quad (3-11)$$

or as the "H-function". The H-function is a M by N by N array. Substituting equation 3-11 into equation 3-10 and arranging for dimensional compatibility with equation 3-8 gives

$$\frac{d}{dt} \left[ G_{\Phi}^u \right] = \dot{\underline{\Phi}}^T \left[ H_{\Phi\Phi}^u \right] \quad (3-12)$$

Substituting equations 3-6 and 3-12 into equation 3-8 gives

$$\ddot{\underline{u}} = [G_{\phi}^u] \ddot{\phi} + \dot{\phi}^T [H_{\phi\phi}^u] \dot{\phi} \quad (3-13)$$

Recalling equations 3-7 and 3-13 and the method of derivation from equation 3-5 to 3-13, as far as possible, the expressions follow the fundamental principle of separating the position(or geometry) dependent terms from the time dependent terms. Also notice that both the G- and H-functions are purely position dependent. Equations 3-7 and 3-13 are the two basic formulations which will be used in this entire work for the velocity and acceleration terms. Freeman and Tesar [14] carried the derivation further to include the third order time derivative of  $\underline{u}$ , which is the jerk. However, this will not be discussed in the present work.

Before going any further, it is important to explicitly define the components that make up the G- and H-functions. Since

$$[G_{\phi}^u] = \frac{\partial \underline{u}}{\partial \phi} \quad (3-6)$$

and

$$[H_{\phi\phi}^u] = \left( \frac{\partial}{\partial \phi} [G_{\phi}^u] \right), \quad (3-11)$$

the G-function can be written as

$$[G_{\phi}^u] = \begin{bmatrix} g_1^1 & g_2^1 & \dots & g_N^1 \\ g_1^2 & & & \\ & & & \\ g_1^M & g_2^M & \dots & g_N^M \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial u^1}{\partial \phi_1} & \frac{\partial u^1}{\partial \phi_2} & \dots & \frac{\partial u^1}{\partial \phi_N} \\ \frac{\partial u^2}{\partial \phi_1} & & & \\ & & & \\ \frac{\partial u^M}{\partial \phi_1} & \frac{\partial u^M}{\partial \phi_2} & \dots & \frac{\partial u^M}{\partial \phi_N} \end{bmatrix} \quad (3-14)$$

where  $g_{\alpha}^m$  is defined as the  $m^{\text{th}}$  row and  $n^{\text{th}}$  column of the  $M$  by  $N$   $[G_{\phi}^u]$  matrix and  $m = 1, 2, \dots, M$  and  $n = 1, 2, \dots, N$ . The  $H$ -function can be written as

$$[H_{\phi\phi}^u]_{i,j,k} = \begin{bmatrix} h_{11}^i & h_{12}^i & \dots & h_{1N}^i \\ h_{21}^i & & & \\ & & & \\ h_{N1}^i & h_{N2}^i & \dots & h_{NN}^i \end{bmatrix}$$

$$= \begin{bmatrix} \frac{\partial^2 u^i}{\partial \phi_1 \partial \phi_1} & \frac{\partial^2 u^i}{\partial \phi_1 \partial \phi_2} & \dots & \frac{\partial^2 u^i}{\partial \phi_1 \partial \phi_N} \\ \frac{\partial^2 u^i}{\partial \phi_2 \partial \phi_1} & & & \\ & & & \\ \frac{\partial^2 u^i}{\partial \phi_N \partial \phi_1} & \frac{\partial^2 u^i}{\partial \phi_N \partial \phi_2} & \dots & \frac{\partial^2 u^i}{\partial \phi_N \partial \phi_N} \end{bmatrix} \quad (3-15)$$

where  $i = 1, 2, \dots, M$ . Note,  $h_{j,k}^i$  is defined as the component of the  $i^{\text{th}}$  plane,  $j^{\text{th}}$  row and  $k^{\text{th}}$  column of the  $M$  by  $N$  by  $N$ ,  $[H_{\phi\phi}^u]$  array.

Until this point, none of the derivations pertain to any particular manipulator. However, they provide the necessary tools for the development of the next few sections and the chapters that follow.

### 3-2.1 Kinematics of Serial Manipulators

The kinematic analyses of serial manipulators has been treated extensively by researchers including Colson and Perreira [5], Craig [7], Freeman and Tesar ([13], [14], [15]), Fu, Gonzalez and Lee [17], Lee [24], Lee and Ziegler [26], Paul [34], Paul, Shimano and Mayer [35], Takano, Yashima and Yada [41], Thomas and Tesar ([42], [43]), and many others. This section is designed specifically for the benefit of those readers who are unfamiliar with the previous mentioned works or for those who are unfamiliar with the notational scheme adopted by Thomas and Tesar ([42], [43]) or Freeman and Tesar ([13], [14], [15]). It is only intended to serve as a brief review of the current topic.

An  $N$  DOF serial manipulator, as the name implies, consists of  $N$  links and  $N$  joints connected together in series (ie., one link after the other). Only two of the lower-pair connectors, revolute and prismatic joints, are considered in this work due to the fact that any other joint can be treated as a combination of the two. A general  $N$  DOF serial manipulator is shown in Fig. 3-1, as adapted from Freeman and Tesar [14]. Each rigid link in the serial chain is characterized by four independent parameters according to the Denavit-Hartenberg convention. They are the link offset ( $S_{ji}$  or  $S_j$ ), joint angle ( $\theta_{ji}$  or  $\theta_j$ ), link length ( $a_{j(j+1)}$ ) and twist angle ( $\alpha_{j(j+1)}$ ). A double subscript here denotes a fixed system parameter, whereas, a single subscript represents a system variable. Following this scheme, a revolute link will have  $S_{ji}$  and  $\theta_j$  as the two independent joint parameters. On the other hand,  $S_j$  and  $\theta_{ji}$  will be the two joint parameters for a prismatic link. The coordinate system adopted here sets the base link (or link 01) coincident with the fixed axes ( $\dot{X}$ ,  $\dot{Y}$ ,  $\dot{Z}$ ), with the first axis of rotation (or translation)  $\dot{Z}_1$  defining the direction of  $\dot{Z}$ .  $\dot{a}^{12}$  is along the fixed axis  $\dot{X}$  when  $\theta_1$  (or  $\theta_{11}$ ) is zero. The superscript with a bracket in  $^{(i)}X$  and  $^{(j)}Z$  represents the local  $X$  and  $Z$  axes.  $R^j$  is

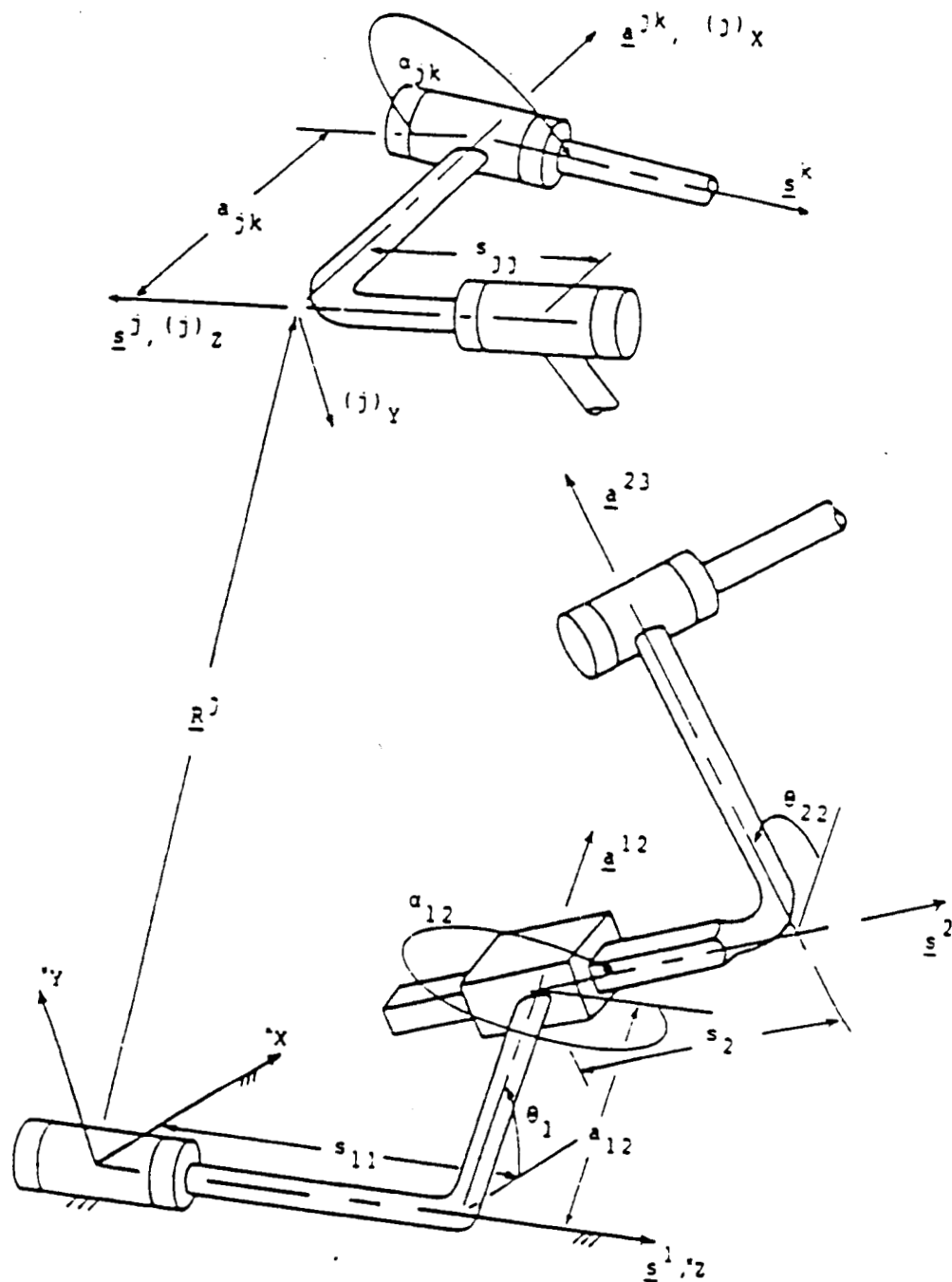


Fig. 3-1 Kinematic Representation of the Serial Manipulator\*

\*Adapted from Freeman and Tesar [14]

the vector pointing from the origin of the fixed reference frame to the origin of the  $j^{\text{th}}$  link frame. This vector can be expressed as a sum of all the preceding vectors  $\mathbf{a}$  and  $\mathbf{z}$  as

$$\mathbf{R}^j = s_{11} \mathbf{z}^1 + \sum_{i=2}^j \left( a_{(i-1)i} \mathbf{a}^{(i-1)i} + s_{ii} \mathbf{z}^i \right) \quad (3-16)$$

Following this convention for setting up the local link frames, the link parameters can be defined as:

- (a) Link offset  $S_{jj}$  or  $S_j$  = the distance from  $\mathbf{a}^{(j-1)j}$  to  $\mathbf{a}^{j(j+1)}$  measured along  $\mathbf{z}^j$  direction;
- (b) Joint angle  $\theta_{jj}$  or  $\theta_j$  = the angle between  $\mathbf{a}^{(j-1)j}$  and  $\mathbf{a}^{j(j+1)}$  measured about  $\mathbf{z}^j$  axis;
- (c) Link length  $a_{j(j+1)}$  = the distance from  $\mathbf{z}^j$  to  $\mathbf{z}^{j+1}$  measured along  $\mathbf{a}^{j(j+1)}$  direction;
- (d) Twist angle  $\alpha_{j(j+1)}$  = the angle between  $\mathbf{z}^j$  and  $\mathbf{z}^{j+1}$  measured about  $\mathbf{a}^{j(j+1)}$  axis.

### 3-2.1.1 First-order Kinematics

General rigid body motion can be conveniently expressed in terms of a translational velocity component and a rotational velocity component. These two velocities can be separated by treating the body as a point to obtain the translational term, and as an object rotating about that point to get the rotational term. Serial linkages are made up of rigid bodies connected together sequentially. In order to determine the velocity at any point in the chain, one needs to first find the angular velocity of the link frame containing the point and then the velocity of the origin of that frame.

*Rotational.* By the angular velocity addition theorem, the angular velocity of link  $jk$  in Fig. 3-1 can be expressed as the sum of all the relative angular velocities of the links preceding link  $jk$ , ie. :

$$\omega^{jk} = \sum_{m=1}^j \dot{\theta}_m \mathbf{z}^m \quad (3-17)$$

where  $\dot{\theta}_m \mathbf{z}^m$  is the relative angular velocity between link  $(m-1)m$  and link  $m(m+1)$  for a revolute joint and equals to zero for a prismatic joint, since  $\dot{\theta}_m$  is zero. Following the technique used to derive equation 3-7, equation 3-17 is separated into the geometric term  $\mathbf{z}^m$  and the time dependent term  $\dot{\theta}_m$ , and expressed as

$$\omega^{jk} = [G_{\phi}^{jk}] \dot{\phi} \quad (3-18)$$

Comparing equations 3-17 and 3-18, it can be shown (Freeman and Tesar [14]) that,

$$[G_{\phi}^{jk}]_{;m} = \begin{cases} \mathbf{z}^m, & m \leq j; \phi_m = \theta_m \text{ for revolute joint} \\ 0, & \text{otherwise} \end{cases} \quad (3-19)$$

Recall in Section 3-1 that  $[G_{\phi}^{jk}]_{;m}$  represents all the rows in column  $m$  of matrix  $[G_{\phi}^{jk}]$ , which is the unit vector  $\mathbf{z}^m$  of joint axis  $m$  expressed in terms of the fixed cartesian reference frame  $(\dot{X}, \dot{Y}, \dot{Z})$ , ie. :

$$\mathbf{z}^m = (\dot{X}^m, \dot{Y}^m, \dot{Z}^m)^T \quad (3-20)$$

There is no angular velocity contribution from prismatic joints, therefore, the zero vector for all other cases, including  $j > m$ .

*Translational.* The translational velocity of a point P fixed in link  $jk$  can be derived by taking the time derivative of the position vector of point P in the fixed reference frame. The position vector of P can be written as :

$${}^j\mathbf{P} = \mathbf{R}^j + [\mathbf{T}^j] {}^{(j)}\mathbf{P} \quad (3-21)$$

where  $\mathbf{R}^j$  is as defined in equation 3-16,  $[\mathbf{T}^j]$  is the rotation matrix relating link frame  $j$  to the fixed coordinate frame. The pre-superscript with a bracket represents a locally referenced vector or component, therefore,  ${}^{(j)}\mathbf{P}$  is the position vector of P with respect to the origin of frame  $j$  and expressed in the local frame coordinates. Taking the time derivative of equation 3-21 gives

$$\dot{{}^j\mathbf{P}} = \dot{\mathbf{s}}_1 \mathbf{s}^1 + \sum_{m=2}^j \left( a_{(m-1)m} \dot{\mathbf{s}}^{(m-1)m} + s_{mm} \dot{\mathbf{s}}^m + \dot{s}_m \mathbf{s}^m \right) + \frac{d}{dt} ([\mathbf{T}^j] {}^{(j)}\mathbf{P}) \quad (3-22)$$

It can be shown (Freeman and Tesar [14]) that simplifying equation 3-22 gives a more compact form as

$$\dot{{}^j\mathbf{P}} = \sum_{m=1}^j \left( \dot{s}_m \mathbf{s}^m + (\dot{\theta}_m \mathbf{s}^m) \times ({}^j\mathbf{P} - \mathbf{R}^m) \right) \quad (3-23)$$

where " $\times$ " denotes a vector cross product. Equation 3-23 gives the necessary form to separate the position dependent terms from the time dependent terms. As a result, equation 3-23 can be expressed in the desired form

$$\dot{{}^j\mathbf{P}} = [\mathbf{G}_\phi^j] \dot{\boldsymbol{\phi}} \quad (3-24)$$

where

$$[{}^jG_\phi^p]_{;m} = \begin{cases} \mathbf{s}^m \times ({}^jP - \mathbf{R}^m), & m \leq j; \phi_m = \theta_m \text{ revolute joint} \\ \mathbf{s}^m, & m \leq j; \phi_m = s_m \text{ prismatic joint} \\ \mathbf{0}, & m > j \end{cases} \quad (3-25)$$

Equation 3-25 simply states that the G-function of joint  $m$  is the vector cross product of its unit vector along the axis of rotation with the vector pointing from the origin of the  $m^{\text{th}}$  link frame to the point  $P$ , if link  $m$  is revolute. In the case of a prismatic joint, the G-function is simply the unit vector along the axis of motion. Otherwise, it is the zero vector (ie.,  $m > j$ ).

### 3-2.1.2 Second-order Kinematics

The reader is referred to the development of equation 3-13 in the beginning of section 3-2, since the derivation of the second-order kinematics here is based heavily on that section.

*Rotational.* The angular acceleration of a rigid body can be obtained by simply differentiating equation 3-18 with respect to time. This yields

$$\begin{aligned} \ddot{\alpha}^{jk} &= \ddot{\omega}^{jk} \\ &= [G_\phi^{jk}] \ddot{\phi} + \left( \frac{d}{dt} [G_\phi^{jk}] \right) \dot{\phi} \end{aligned} \quad (3-26)$$

Recalling equation 3-12

$$\frac{d}{dt} [G_\phi^u] = \dot{\phi}^T [H_{\phi\phi}^u] \quad (3-12)$$

The second term in bracket in equation 3-26 can then be expressed as

$$\frac{d}{dt} [G_\phi^{jk}] = \dot{\phi}^T [H_{\phi\phi}^{jk}] \quad (3-27)$$

Substituting equation 3-27 into equation 3-26 gives

$$\ddot{\alpha}^{jk} = [G_{\varphi}^{jk}] \ddot{\varphi} + \dot{\varphi}^T [H_{\varphi\varphi}^{jk}] \dot{\varphi} \quad (3-28)$$

where  $[H_{\varphi\varphi}^{jk}]$  is a 3 by M by M array with its  $i^{\text{th}}$  plane corresponding to the  $i^{\text{th}}$  row of the vector  $\alpha^{jk}$ .

Recall equation 3-19 for the rotational G-function. Taking the time derivative of equation 3-19 results in

$$\frac{d}{dt} [G_{\varphi}^{jk}]_{;m} = \begin{cases} \dot{s}^m, & m \leq j; \varphi_m = \theta_m \text{ for revolute joint} \\ 0, & \text{otherwise} \end{cases} \quad (3-29)$$

Since

$$\dot{s}^m = \left( \sum_{i=1}^{m-1} \dot{\theta}_i s^i \right) \times s^m \quad (3-30)$$

using the relationship

$$\frac{\partial}{\partial \varphi_i} [G_{\varphi}^{jk}]_{;m} = \frac{\partial}{\partial \varphi_i} \left( \frac{d}{dt} [G_{\varphi}^{jk}]_{;m} \right) \quad (3-31)$$

the non-symmetric rotational second-order influence coefficients are

$$[H_{\varphi\varphi}^{jk}]_{;m;n} = \begin{cases} s^m \times s^n; & m < n \leq j; \varphi_m = \theta_m, \varphi_n = \theta_n \\ 0 & ; \text{otherwise} \end{cases} \quad (3-32)$$

$[H_{\varphi\varphi}^{jk}]_{;m;n}$  is the vector component of the array  $[H_{\varphi\varphi}^{jk}]$ , which can be viewed as a 3 by 1 vector running into the three planes of  $[H_{\varphi\varphi}^{jk}]$  at  $m^{\text{th}}$  row and  $n^{\text{th}}$  column of the planes. The  $i^{\text{th}}$  row in  $[H_{\varphi\varphi}^{jk}]_{;m;n}$  corresponds to the  $i^{\text{th}}$  plane in  $[H_{\varphi\varphi}^{jk}]$ .

*Translational.* The derivation for the translational acceleration of a rigid body is more involved than the angular acceleration described above. This author will not give the detailed derivation but rather just the final result. Readers

who are interested in the derivation are urged to refer to Freeman and Tesar [14]. However, the general procedure is still similar to that of the angular acceleration. By taking the time derivative of equation 3-24, the result is

$${}^i\ddot{\mathbf{P}} = [{}^i\mathbf{G}_\phi^p] \ddot{\boldsymbol{\Phi}} + \left( \frac{d}{dt} [{}^i\mathbf{G}_\phi^p] \right) \dot{\boldsymbol{\Phi}} \quad (3-33)$$

Using the general result of equation 3-27 gives

$${}^i\ddot{\mathbf{P}} = [{}^i\mathbf{G}_\phi^p] \ddot{\boldsymbol{\Phi}} + \dot{\boldsymbol{\Phi}}^T [{}^i\mathbf{H}_{\phi\phi}^p] \dot{\boldsymbol{\Phi}} \quad (3-34)$$

where

$$[{}^i\mathbf{H}_{\phi\phi}^p]_{m;n} = \begin{cases} \mathbf{S}^m \times \left[ \mathbf{S}^n \times ({}^i\mathbf{P} - \mathbf{R}^n) \right] & , m < n \leq j ; R R \\ \mathbf{S}^n \times \left[ \mathbf{S}^m \times ({}^i\mathbf{P} - \mathbf{R}^m) \right] & , n < m \leq j ; R R \\ \mathbf{S}^n \times \mathbf{S}^m & , n < m \leq j ; P R \\ \mathbf{S}^m \times \mathbf{S}^n & , m < n \leq j ; R P \\ Q & , m < n \leq j ; P R \\ Q & , n < m \leq j ; R P \\ Q & , m \text{ or } n > j ; \text{all} \end{cases} \quad (3-35)$$

The first capital letter at the end of each row represents the joint type of  $m$ , the second one represents the joint type of  $n$ ;  $R$  stands for revolute joint and  $P$  stands for prismatic joint. Notice that the form of equation 3-35 gives a symmetric matrix for each plane of  $[{}^i\mathbf{H}_{\phi\phi}^p]$ , which is different from the non-symmetric planes of the second-order rotational influence coefficients. The results for all the  $G$ - and  $H$ -functions are listed in Table 3-3 and Table 3-4, respectively.

Symbol	Joint Type		Restriction	Value
	m	n		
Rotational $[G_{\phi}^{jk}]_{;m}$	-	-	$m \leq j$	$\underline{s}^m$
	-	-	$m > j$	$\underline{0}$
	-	-	All m	$\underline{0}$
Translational $[{}^jG_{\phi}^p]_{;m}$	-	R	$m \leq j$	$\underline{s}^m \times ({}^j\underline{P} - \underline{R}^m)$
	-	P	$m \leq j$	$\underline{s}^m$
	-	R,P	$m > j$	$\underline{0}$

**Table 3-3 First-order Kinematic Influence Coefficients for Serial Manipulators**

Symbol	<u>Joint Type</u>		Restriction	Value
	m	n		
Rotational $[H_{\phi\phi}^{jk}]_{m;n}$	R	R	$m < n \leq j$	$\underline{s}^m \times \underline{s}^n$
	R	R	$m < n$ or $n > j$	0
	P or R	P	All m, n	0
Translational $[{}^jH_{\phi\phi}^p]_{m;n}$	R	R	$m < n \leq j$	$\underline{s}^m \times \left[ \underline{s}^n \times \left( {}^jP - R^n \right) \right]$
	R	R	$n < m \leq j$	$\underline{s}^n \times \left[ \underline{s}^m \times \left( {}^jP - R^m \right) \right]$
	P	R	$n < m \leq j$	$\underline{s}^n \times \underline{s}^m$
	R	P	$m < n \leq j$	$\underline{s}^m \times \underline{s}^n$
	P	R	$m < n \leq j$	0
	R	P	$n < m \leq j$	0
	All cases		m or n > j	0

Table 3-4 Second-order Kinematic Influence Coefficient for Serial Manipulators

### 3-2.1.3 Forward Kinematics

After determining the first- and second-order kinematic influence coefficients of a serial manipulator, we are ready to investigate the forward kinematics (ie., position, velocity and acceleration) of serial manipulators. In a practical situation, rather than specifying the state of the joints, often called the joint space, usually the kinematic state of the end-effector, which is often called the Cartesian space, is specified. It is frequently trivial to solve for the forward kinematic state of any manipulator. Given the kinematic state of each of the joints, the position and orientation of the end-effector can easily be determined by using any of the fundamental geometric or algebraic approaches (Craig [7], Fu, Gonzalez and Lee [17], Paul [34]). This will not be discussed in this work. However, in order to aid those readers who are unfamiliar with the use of the G- and H-functions, the forward kinematic analysis for velocity and acceleration will be briefly reviewed here serving as a prelude to the reverse kinematics that will be discussed in the next section.

Consider a general six DOF serial manipulator with a point P fixed in the last link (or the end-effector). Knowing the kinematic state of each of the joints, represented by the symbol  $\Phi$  as the generalized joint inputs, the kinematic state of the end-effector, represented by  $\underline{u}$ , can then be determined.  $\Phi$  is a 6 by 1 vector for a six DOF manipulator, and  $\underline{u}$  is always a 6 by 1 vector. The resulting forward kinematic model is

$$\dot{\underline{u}} = \left[ G_{\Phi}^u \right] \dot{\Phi} \quad (3-36)$$

and

$$\ddot{\underline{u}} = \left[ G_{\Phi}^u \right] \ddot{\Phi} + \dot{\Phi}^T \left[ H_{\Phi\Phi}^u \right] \dot{\Phi} \quad (3-37)$$

where the first three rows of  $\dot{\underline{u}}$  and  $\ddot{\underline{u}}$  are the translational terms, and the last three rows are the angular terms. The combined 6 by 6 G-function is then given by

$$[G_{\phi}^u] = \begin{bmatrix} [{}^6G_{\phi}^p] \\ [G_{\phi}^{67}] \end{bmatrix} \quad (3-38)$$

where the top consists of the 3 by 6 translational G-functions and the bottom contains the 3 by 6 rotational G-functions. The H-function is formed as

$$[H_{\phi\phi}^u]_{i,:} = [{}^6H_{\phi\phi}^p]_{i,:} ; i = 1, 2, 3 \quad (3-39)$$

and

$$[H_{\phi\phi}^u]_{i,:} = [H_{\phi\phi}^{67}]_{i,:} ; i = 4, 5, 6 \quad (3-40)$$

The combination of equations 3-39 and 3-40 gives a 6 by 6 by 6 array with its first three planes (equation 3-39) corresponding to the first three rows of  $\ddot{\mathbf{u}}$  for the translational acceleration  $\ddot{\mathbf{P}}^6$ , and the last three planes (equation 3-40) corresponding to the last three rows of  $\ddot{\mathbf{u}}$  for the angular acceleration  $\ddot{\alpha}^{67}$ .

#### 3-2.1.4 Reverse Kinematics

Having been introduced to the application of the G- and H-functions for the forward kinematics, this section will discuss the more operationally difficult problem of reverse kinematics. This involves much more complex solutions and physical considerations. Before solving the reverse kinematics problem, it will be beneficial to discuss the complexity of the solution(s).

There are three main concerns with the reverse kinematic solutions (Craig [7]). The first concern is the existence of solutions. Do the solutions exist? This raises the question of manipulator workspace. There are two kinds of manipulator workspaces. One is called the dextrous workspace, which is defined as the volume of space within which the end-effector can reach in all directions. The other is called the reachable workspace, which is defined as the volume of space that the robot can reach in at least one direction. Hence, the specification of the end-effector must be within the possible workspace of the robot.

The second concern is the problem of multiple solutions. Unlike forward kinematics which yield a unique solution, reverse kinematics can have

more than one solution with the same end-effector specification. This requires the selection of one of the possible solutions. The robot controller must be capable of making this kind of decision. For example, to go from point A to point B in Fig. 3-2, the robot has the choice of two configurations, I and II. The robot has to decide which is the more efficient way to take and if there is an obstacle, which is the best way to avoid it.

Being able to resolve the previous two concerns still does not guarantee a satisfactory solution. There is the problem of singularity. If the specification for the end-effector puts the manipulator in a singularity configuration, the robot must be able to foresee the problem and work out an acceptable alternate solution.

Knowing the problems related to reverse kinematics, the next discussion will concentrate on deriving the solution(s). Given the end-effector states ( $\underline{u}$ ,  $\dot{\underline{u}}$  and  $\ddot{\underline{u}}$ ), it is desired to determine the state of all the joints ( $\underline{\phi}$ ,  $\dot{\underline{\phi}}$  and  $\ddot{\underline{\phi}}$ ), where

$$\phi = \theta \quad ; \quad \text{for revolute joint}$$

and

$$\phi = S \quad ; \quad \text{for prismatic joint}$$

From equations 3-36 and 3-37, we can arrive at the following equations

$$\begin{aligned} \dot{\underline{\phi}} &= [G_{\phi}^u]^{-1} \dot{\underline{u}} \\ &= [G_{\phi}^u] \dot{\underline{u}} \end{aligned} \quad (3-41)$$

$$\ddot{\underline{\phi}} = [G_{\phi}^u]^{-1} \left( \ddot{\underline{u}} - \dot{\underline{\phi}}^T [H_{\phi\phi}^u] \dot{\underline{\phi}} \right) \quad (3-42)$$

Substitute equation 3-41 into equation 3-42 gives

$$\ddot{\underline{\phi}} = [G_{\phi}^u]^{-1} \ddot{\underline{u}} - [G_{\phi}^u]^{-1} \left( \dot{\underline{u}}^T [G_{\phi}^u]^T [H_{\phi\phi}^u] [G_{\phi}^u]^{-1} \dot{\underline{u}} \right) \quad (3-43)$$



Using the method of generalized dot product(Appendix A), defined by Freeman and Tesar[14], the second-order reverse kinematic state is given by

$$\begin{aligned}\ddot{\Phi} &= [G_{\Phi}^u]^{-1} \ddot{u} - \dot{u} [G_{\Phi}^u]^{-T} \left( [G_{\Phi}^u]^{-1} \cdot [H_{\Phi\Phi}^u] \right) [G_{\Phi}^u]^{-1} \dot{u} \\ &= [G_{\Phi}^u]^{-1} \ddot{u} + \dot{u} [H_{\Phi u}^u] \dot{u}\end{aligned}\quad (3-44)$$

However, there is still one serious problem, and informed readers should ponder for a minute and ask the question how to determine  $[G_{\Phi}^u]$  and  $[H_{\Phi\Phi}^u]$ . Recall in Section 3-2.1.3,  $[G_{\Phi}^u]$  and  $[H_{\Phi\Phi}^u]$  are functions of the joint inputs  $\Phi$ . Thus, before arriving at equations 3-41 and 3-44,  $\Phi$  has to be found. There are many different approaches which yield solutions for the joint inputs given the end-effector position and orientation. Interested readers are strongly urged to refer to Duffy [11] as reverse position analysis will not be discussed here.

Equations 3-41 and 3-44 are most useful in the context of this work, of course, provided that  $[G_{\Phi}^u]$  and  $[H_{\Phi\Phi}^u]$  are already known. The two equations give a systematic and general approach to solving the first- and second-order geometric(and time) derivatives of the joint parameters( $\Phi$ ) in terms of the end-effector parameters( $u$ ), which are among the most essential elements in determining the cartesian based robot dynamic model. The first- and second-order KIC for serial manipulators with revolute and prismatic links are listed in Table 3-3 and Table 3-4. Freeman and Tesar [14] expand the derivation to third-order KIC, which again will not be discussed here.

### 3-3 General Approach to Develop the Dynamic Model of Serial Manipulators

So far this investigation has focused on the study of kinematic considerations(ie., velocity and acceleration) of serial manipulator. Nothing has been said about the generalized forces or torques associated with the rigid body motion. The determination of the forces or torques that are necessary to cause the

desired motion is another essential question to be answered in studying any manipulator (Greenwood [19], Hollerbach [20], Kane and Levinson [23], Lee, Lee and Nigam [25], Takano, Yashima and Yada [41], Torby [44], Tourassis and Neuman ([45], [46]), Walker and Orin [47]). This section will discuss the development of the dynamic model of serial manipulators.

The derivation of the manipulator dynamics in this work is based almost entirely on Freeman and Tesar [14]. The two fundamental principles of mechanics employed in the derivation are the principle of virtual work and d'Alembert's principle. Similar to the kinematic model described in the previous section, the form of the dynamic equations is also expressed in terms of configuration dependent and time dependent terms. This separation greatly facilitates the transfer of generalized coordinates employed in developing the dynamic model.

The dynamic model developed will be separated into two parts. The first part is the effect of applied loads on the manipulator generalized input loads by using the principle of virtual work. Then, d'Alembert's principle will be applied to study the effect of the manipulator's inertia. The combination of these two parts enables one to express a highly geometric form of the dynamic model for a manipulator.

### 3-3.1 Applied Loads

Noting the simplicity of the robot kinematic expressions that result from separating the rotational and translational KIC, the derivation of the dynamic model of serial manipulators will follow this preferred scheme. Consider a  $M$  DOF serial manipulator in space with a set of  $M$ , 3 by 1 force vectors  ${}^j\mathbf{f}^P$  along with a set of  $M$ , 3 by 1 moment vectors  ${}^j\mathbf{m}^{jk}$  ( $j=1, 2, \dots, M$  and  $k=j+1$ ), acting respectively on point  ${}^jP$  and link  $jk$ . Using the principle of virtual work, the set of  $M$  generalized input loads required to offset the applied loads and keep the manipulator in static equilibrium can be determined. The virtual work done by the applied loads is

$$\delta W^L = \sum_{j=1}^M \left\{ \left( {}^j \mathbf{f}^p \right)^T \left[ {}^j \mathbf{G}_\phi^p \right] + \left( \mathbf{m}^{jk} \right)^T \left[ \mathbf{G}_\phi^{jk} \right] \right\} \dot{\phi} \delta t \quad (3-45)$$

and by the input loads is

$$\delta W_\phi = \left( \mathbf{I}_\phi \right)^T \dot{\phi} \delta t \quad (3-46)$$

According to the principle of virtual work the total virtual work done on the system, which must be zero for equilibrium, is

$$\delta W = \delta W^L + \delta W_\phi = 0 \quad (3-47)$$

or, substituting equations 3-45 and 3-46 into equation 3-47,

$$\delta W = \left\{ \sum_{j=1}^M \left\{ \left( {}^j \mathbf{f}^p \right)^T \left[ {}^j \mathbf{G}_\phi^p \right] + \left( \mathbf{m}^{jk} \right)^T \left[ \mathbf{G}_\phi^{jk} \right] \right\} + \left( \mathbf{I}_\phi \right)^T \right\} \dot{\phi} \delta t = 0 \quad (3-48)$$

Equation 3-48 implies that

$$\mathbf{I}_\phi^L = \sum_{j=1}^M \left\{ \left[ {}^j \mathbf{G}_\phi^p \right]^T {}^j \mathbf{f}^p + \left[ \mathbf{G}_\phi^{jk} \right]^T \mathbf{m}^{jk} \right\} \quad (3-49)$$

where  $\mathbf{I}_\phi^L$  is defined as the effective input loads. Notice that the effective input load  $\mathbf{I}_\phi^L$  has opposite sign from the required offset input loads  $\mathbf{I}_\phi$ . The G- and H- functions are as defined by equations 3-19 and 3-25.

### 3-3.2 Inertial Loads

Consider the same M DOF serial manipulator as in Section 3-3.1. Defining the locally referenced inertia tensor as  $\begin{bmatrix} {}^{(j)} \Pi^{jk} \end{bmatrix}$ , the local angular momentum of link  $jk$  is

$${}^{(j)} \mathbf{L}^{jk} = \begin{bmatrix} {}^{(j)} \Pi^{jk} \end{bmatrix} {}^{(j)} \dot{\phi}^{jk} \quad (3-50)$$

where  ${}^{(j)}\dot{\omega}^{jk}$  is the absolute angular velocity of link  $jk$  expressed in terms of the local link frame. To express equation 3-50 in the fixed referenced frame, simply premultiply the equation by a rotation matrix  $[T^j]$ . This gives

$$\underline{L}^{jk} = [T^j] {}^{(j)}\underline{L}^{jk} = [T^j] \left[ {}^{(j)}\Pi^{jk} \right] {}^{(j)}\dot{\omega}^{jk} \quad (3-51)$$

Since

$$\dot{\omega}^{jk} = [T^j] {}^{(j)}\dot{\omega}^{jk} \quad (3-52)$$

rewriting equation 3-52 gives

$${}^{(j)}\dot{\omega}^{jk} = [T^j]^{-1} \dot{\omega}^{jk} \quad (3-53)$$

Also, the inverse of the rotation matrix  $[T^j]$  is its transpose due to its orthonormal property. Therefore, substituting equation 3-53 into equation 3-51 gives

$$\underline{L}^{jk} = [T^j] \left[ {}^{(j)}\Pi^{jk} \right] [T^j]^T \dot{\omega}^{jk} \quad (3-54)$$

Notice that the angular momentum is now expressed in the fixed reference frame. By defining the globally referenced inertia tensor as

$$[\Pi^{jk}] = [T^j] \left[ {}^{(j)}\Pi^{jk} \right] [T^j]^T \quad (3-55)$$

equation 3-55 can be written as

$$\underline{L}^{jk} = [\Pi^{jk}] \dot{\omega}^{jk} \quad (3-56)$$

Before continuing with the inertial load derivation, there is one more term that needs attention. It is the mass center of each link, which is located by

$${}^{jk}\underline{C} = \underline{R}^j + [T^j]^T {}^{(j)}\underline{C} \quad (3-57)$$

where  $\mathbf{R}^j$  is the vector from the origin of the fixed reference frame to the origin of link  $jk$  local reference frame.  ${}^{(i)}\mathbf{C}$  is the vector pointing from the local origin to the center of mass expressed in the local frame. The linear velocity and acceleration of the link center can be found by taking the time derivative of equation 3-57. The results are in the same form as equations 3-24 and 3-34 for the velocity and acceleration, respectively, due to the similarity between equations 3-21 and 3-57. This gives the velocity of the center of mass as

$${}^{jk}\dot{\mathbf{C}} = \left[ {}^j\mathbf{G}_{\phi}^c \right] \dot{\phi} \quad (3-58)$$

and the acceleration as

$${}^{jk}\ddot{\mathbf{C}} = \left[ {}^j\mathbf{G}_{\phi}^c \right] \ddot{\phi} + \dot{\phi}^T \left[ {}^j\mathbf{H}_{\phi}^c \right] \dot{\phi} \quad (3-59)$$

Finally, applying the generalized principle of d'Alembert, which is the combination of the principle of virtual work and the principle of d'Alembert, together with the Newton-Euler equations of motion gives the total effective inertial load as

$$\mathbf{I}_{\phi}^I = \sum_{j=1}^M \left\{ \left[ {}^j\mathbf{G}_{\phi}^c \right]^T M_{jk} {}^{jk}\ddot{\mathbf{C}} + \left[ \mathbf{G}_{\phi}^{jk} \right]^T \left( \left[ \Pi^{jk} \right] \underline{\alpha}^{jk} + \underline{\omega}^{jk} \times \left( \left[ \Pi^{jk} \right] \underline{\omega}^{jk} \right) \right) \right\} \quad (3-60)$$

Equation 3-60 correctly describes the effective inertial loads, but it is not in the desired form. Recall the fundamental rule in this work is to separate the position(or configuration) dependent terms from the time dependent terms. Freeman and Tesar [14] showed that the last term in equation 3-60 can be expressed without the cross product (in a quadratic form) as

$$\begin{aligned}
{}^{jk}\underline{\omega} \times \left( [{}^{jk}\Pi] {}^{jk}\underline{\omega} \right) &= \left( {}^{jk}\underline{\omega} \right)^T [{}^{jk}\Xi] {}^{jk}\underline{\omega} \\
&= \begin{bmatrix} \left( {}^{jk}\underline{\omega} \right)^T [{}^{jk}\Xi]_{1;;} {}^{jk}\underline{\omega} \\ \left( {}^{jk}\underline{\omega} \right)^T [{}^{jk}\Xi]_{2;;} {}^{jk}\underline{\omega} \\ \left( {}^{jk}\underline{\omega} \right)^T [{}^{jk}\Xi]_{3;;} {}^{jk}\underline{\omega} \end{bmatrix}
\end{aligned} \tag{3-61}$$

where

$$\begin{aligned}
[{}^{jk}\Xi]_{1;;} &= \begin{bmatrix} \underline{0} & | & [{}^{jk}\Pi]_{3;}^T & | & -[{}^{jk}\Pi]_{2;}^T \end{bmatrix} \\
[{}^{jk}\Xi]_{2;;} &= \begin{bmatrix} [{}^{jk}\Pi]_{3;}^T & | & \underline{0} & | & -[{}^{jk}\Pi]_{1;}^T \end{bmatrix} \\
[{}^{jk}\Xi]_{3;;} &= \begin{bmatrix} [{}^{jk}\Pi]_{2;}^T & | & -[{}^{jk}\Pi]_{1;}^T & | & \underline{0} \end{bmatrix}
\end{aligned} \tag{3-62}$$

Now, substituting equation 3-28 for  $\underline{Q}^{jk}$  and equation 3-18 for  $\underline{\omega}^{jk}$  along with equation 3-61 into equation 3-60 gives

$$\begin{aligned}
\mathbf{I}_{\Phi}^I &= \sum_{j=1}^M \left\{ [{}^jG_{\Phi}^c]^T M_{jk} \left( [{}^jG_{\Phi}^c] \ddot{\underline{\Phi}} + \dot{\underline{\Phi}}^T [{}^jH_{\Phi}^c] \dot{\underline{\Phi}} \right) \right. \\
&\quad + [G_{\Phi}^{jk}]^T [{}^{jk}\Pi] \left( [G_{\Phi}^{jk}] \ddot{\underline{\Phi}} + \dot{\underline{\Phi}}^T [H_{\Phi}^{jk}] \dot{\underline{\Phi}} \right) \\
&\quad \left. + [G_{\Phi}^{jk}]^T \left( \dot{\underline{\Phi}}^T [G_{\Phi}^{jk}]^T \right) [{}^{jk}\Xi] \left( [G_{\Phi}^{jk}] \dot{\underline{\Phi}} \right) \right\}
\end{aligned} \tag{3-63}$$

Rearranging gives

$$\begin{aligned}
\mathbf{I}_\phi^I = & \sum_{j=1}^M \left\{ \mathbf{M}_{jk} \left[ {}^j\mathbf{G}_\phi^c \right]^T \left[ {}^j\mathbf{G}_\phi^c \right] + \left[ \mathbf{G}_\phi^{jk} \right]^T \left[ \Pi^{jk} \right] \left[ \mathbf{G}_\phi^{jk} \right] \right\} \ddot{\phi} \\
& + \sum_{j=1}^M \left\{ \left( \mathbf{M}_{jk} \left[ {}^j\mathbf{G}_\phi^c \right]^T \dot{\phi}^T \left[ {}^j\mathbf{H}_{\phi\phi}^c \right] \dot{\phi} \right) + \left( \left[ \mathbf{G}_\phi^{jk} \right]^T \left[ \Pi^{jk} \right] \dot{\phi}^T \left[ \mathbf{H}_{\phi\phi}^{jk} \right] \dot{\phi} \right) \right. \\
& \left. + \left[ \mathbf{G}_\phi^{jk} \right]^T \left( \dot{\phi}^T \left[ \mathbf{G}_\phi^{jk} \right]^T \right) \left[ \Xi^{jk} \right] \left( \left[ \mathbf{G}_\phi^{jk} \right] \dot{\phi} \right) \right\}
\end{aligned} \tag{3-64}$$

Utilizing the generalized dot product mentioned in the previous section, equation 3-64 can then be expressed as

$$\mathbf{I}_\phi^I = \left[ \dot{\mathbf{I}}_{\phi\phi} \right] \ddot{\phi} + \dot{\phi}^T \left[ \mathbf{P}_{\phi\phi\phi} \right] \dot{\phi} \tag{3-65}$$

where

$$\left[ \dot{\mathbf{I}}_{\phi\phi} \right] = \sum_{j=1}^M \left\{ \mathbf{M}_{jk} \left[ {}^j\mathbf{G}_\phi^c \right]^T \left[ {}^j\mathbf{G}_\phi^c \right] + \left[ \mathbf{G}_\phi^{jk} \right]^T \left[ \Pi^{jk} \right] \left[ \mathbf{G}_\phi^{jk} \right] \right\} \tag{3-66}$$

and

$$\begin{aligned}
\left[ \mathbf{P}_{\phi\phi\phi} \right] = & \sum_{j=1}^M \left\{ \mathbf{M}_{jk} \left[ {}^j\mathbf{G}_\phi^c \right]^T \cdot \left[ {}^j\mathbf{H}_{\phi\phi}^c \right] + \left[ \mathbf{G}_\phi^{jk} \right]^T \left[ \Pi^{jk} \right] \cdot \left[ \mathbf{H}_{\phi\phi}^{jk} \right] \right. \\
& \left. + \left[ \mathbf{G}_\phi^{jk} \right]^T \left( \left[ \mathbf{G}_\phi^{jk} \right]^T \cdot \left[ \Xi^{jk} \right] \right) \left[ \mathbf{G}_\phi^{jk} \right] \right\}
\end{aligned} \tag{3-67}$$

Notice that equation 3-65 continues to separate the configuration dependent terms from the time dependent terms. The form of equation 3-65 is most desirable for the dynamic model of the effective inertial load of serial manipulators, in this context.

### 3-3.3 The Dynamic Equations

Having determined the equations for the applied and effective inertial loads, the dynamic equations for a general M DOF serial manipulator can then be expressed as

$$\begin{aligned} \mathbf{I}_\phi &= \mathbf{I}_\phi^I - \mathbf{I}_\phi^L \\ &= [\dot{\mathbf{I}}_{\phi\phi}] \ddot{\phi} + \dot{\phi}^T [\dot{\mathbf{P}}_{\phi\phi\phi}] \dot{\phi} - \sum_{j=1}^M \left\{ \left[ {}^j\mathbf{G}_\phi^p \right]^T {}^j\mathbf{f}^p + \left[ \mathbf{G}_\phi^{jk} \right]^T \mathbf{m}^{jk} \right\} \end{aligned} \quad (3-68)$$

Equation 3-68 is expressed in a very compact and explicit form, yet not losing the fundamental rule of separating the configuration dependent terms from the time dependent terms. As a result of this formulation, modeling a serial manipulator can be broken down into the following steps:

- (a) set up the coordinate frame for each of the links according to the convention adopted in Section 3-2.1;
- (b) specify the desired position and orientation of the end-effector;
- (c) perform the reverse position analysis to determine all the joint variables ( $\theta_i$  for revolute joint and  $S_i$  for prismatic joint), and the location of the center of mass for each link;
- (d) if necessary, multiply any of the expressions by the rotation matrix  $[\mathbf{T}^j]$  to express it in terms of the global coordinate frame;
- (e) form the G- and H-functions for the manipulator including the G- and H-functions for the mass center of each link, using Table 3-3 and Table 3-4;
- (f) form  $[\dot{\mathbf{I}}_{\phi\phi}]$  and  $[\dot{\mathbf{P}}_{\phi\phi\phi}]$ ;
- (g) substitute into equation 3-68 for the dynamic equation.

The simulation form of the dynamic equation can easily be obtained by rewriting equation 3-65 into the following form

$$\ddot{\Phi} = [\dot{I}_{\Phi\Phi}]^{-1} \left\{ I_{\Phi} - \dot{\Phi}^T [P_{\Phi\Phi\Phi}] \dot{\Phi} + \sum_{j=1}^M \left( [G_{\Phi}^{jk}]^T \mathbf{m}^{jk} + [{}^j G_{\Phi}^p]^T {}^j \mathbf{f}^p \right) \right\} \quad (3-69)$$

Integration of equation 3-69 will give the time history of the joint parameters resulting from the application of the generalized input loads. This concludes the discussion on the development of the kinematic and dynamic model of serial manipulators using the method of KIC. For more detail, readers are referred to Freeman and Tesar [14]. The next chapter will expand the modeling technique to include parallel manipulators via the transfer of generalized coordinates.

## CHAPTER 4

### TRANSFER OF GENERALIZED COORDINATES

The discussion thus far has concentrated on finding the kinematic and dynamic models of serial manipulators in terms of the relative joint parameters. However, these may not necessarily be the models that one is interested in. For design and control purposes, one may be interested in having the system model referenced to a different set of generalized coordinates. In the case of parallel mechanisms, one may desire to have the system model referenced to a common set of generalized coordinates. Unfortunately, at times, this model may be impractical, if not impossible, to obtain directly. But, as it is usually true, most mechanisms can be modeled in terms of at least one set of generalized coordinates directly. This leaves researchers with the alternative to first determine the easily, or at least practically possible, direct model, then find a way to arrive at the desired model via the direct model.

Freeman and Tesar [14] proposed using an isomorphic transformation technique called the transfer of generalized coordinates to obtain the desired model from the direct models. The development of this technique is based almost entirely on the principle of virtual work. This transfer of generalized coordinates procedure gives the method of KIC more generality and potential. As a result of this development, the method of KIC can now be used to model redundant or overconstrained systems, dual or multi-arms robots, and above all, multi-loop parallel mechanisms which is the theme of this work.

#### 4-1 Kinematic Model Transfer

The transfer of the kinematic model can be broken down into two parts. One part is called the direct kinematic model transfer which has been briefly discussed in Section 3-2.1.4. And obviously, the other part is the indirect kinematic model transfer. Direct kinematic model transfer is defined as any transformation that involves only the interchanging of dependent and independent system parameters.

On the other hand, indirect kinematic model transfer involves the introduction of one or more intermediate coordinate sets before arriving at the desired model.

#### 4-1.1 Direct Kinematic Model Transfer

Consider again the resulting forward kinematic model given in Section 3-2.1.3. The first- and second-order time derivatives of a set of general position parameters( $\underline{u}$ ) are given as in equations 3-36 and 3-37,

$$\dot{\underline{u}} = \left[ G_{\varphi}^u \right] \dot{\underline{\varphi}} \quad (3-36)$$

$$\ddot{\underline{u}} = \left[ G_{\varphi}^u \right] \ddot{\underline{\varphi}} + \dot{\underline{\varphi}}^T \left[ H_{\varphi\varphi}^u \right] \dot{\underline{\varphi}} \quad (3-37)$$

For the purpose of showing that  $\underline{u}$  need not always represent the kinematic state of the end-effector, instead of using  $\underline{u}$  as defined in Chapter 3,  $\underline{q}$  will be used to represent the dependent system parameter, and  $\underline{\varphi}$  will represent any independent set of generalized coordinates, in the initial model. Following this scheme, the initial kinematic model is

$$\dot{\underline{q}} = \left[ G_{\varphi}^q \right] \dot{\underline{\varphi}} \quad (4-1)$$

and

$$\ddot{\underline{q}} = \left[ G_{\varphi}^q \right] \ddot{\underline{\varphi}} + \dot{\underline{\varphi}}^T \left[ H_{\varphi\varphi}^q \right] \dot{\underline{\varphi}} \quad (4-2)$$

However, the desired kinematic model is to have  $\underline{q}$  as the independent system parameter set, and  $\underline{\varphi}$  as the dependent system parameter set. This gives the form for the kinematic model as

$$\dot{\underline{\varphi}} = \left[ G_q^{\varphi} \right] \dot{\underline{q}} \quad (4-3)$$

and

$$\ddot{\underline{\varphi}} = \left[ G_q^{\varphi} \right] \ddot{\underline{q}} + \dot{\underline{q}}^T \left[ H_{qq}^{\varphi} \right] \dot{\underline{q}} \quad (4-4)$$

Rearranging equations 4-1 and 4-2 also can give a form similar to equations 4-3 and 4-4

$$\dot{\Phi} = [G_{\Phi}^q]^{-1} \dot{q} \quad (4-5)$$

and

$$\ddot{\Phi} = [G_{\Phi}^q]^{-1} \ddot{q} + [G_{\Phi}^q]^{-1} \dot{\Phi}^T [H_{\Phi\Phi}^q] \dot{\Phi} \quad (4-6)$$

Substituting equation 4-5 into equation 4-6 for  $\ddot{\Phi}$  gives

$$\ddot{\Phi} = [G_{\Phi}^q]^{-1} \ddot{q} - [G_{\Phi}^q]^{-1} \left( \dot{q}^T [G_{\Phi}^q]^{-T} \right) [H_{\Phi\Phi}^q] \left( [G_{\Phi}^q]^{-1} \dot{q} \right) \quad (4-7)$$

Utilizing the generalized dot product introduced in Chapter 3 gives

$$\ddot{\Phi} = [G_{\Phi}^q]^{-1} \ddot{q} - \dot{q}^T [G_{\Phi}^q]^{-T} \left( [G_{\Phi}^q]^{-1} \cdot [H_{\Phi\Phi}^q] \right) [G_{\Phi}^q]^{-1} \dot{q} \quad (4-8)$$

Comparing equation 4-3 with equation 4-5, it is seen that

$$[G_q^{\Phi}] = [G_{\Phi}^q]^{-1} \quad (4-9)$$

Similarly, comparison of equation 4-4 and equation 4-7 gives

$$[H_{qq}^{\Phi}] = -[G_{\Phi}^q]^{-T} \left( [G_{\Phi}^q]^{-1} \cdot [H_{\Phi\Phi}^q] \right) [G_{\Phi}^q]^{-1} \quad (4-10)$$

Equations 4-9 and 4-10 show how G- and H-functions change when the dependent system parameter becomes the independent system parameter, and vice versa. Notice that this interchange is possible only if  $[G_{\Phi}^q]$  is invertable, which means that it is non-singular and that  $q$  must be a potential set of generalized coordinates.

As mentioned in Chapter 2, at singularity configurations, serial manipulators lose one or more DOF while parallel manipulators lose control of one or more DOF. However, for the purpose of this discussion, it is always assumed that  $[G_{\Phi}^q]$  is non-singular. Also realize how the post super- and subscript help to

denote the dependent and independent system parameters while performing the transformation. They also serve as a checking device when writing the equations. Any error will show up as an irregularity in the form of the super- and subscripts.

#### 4-1.2 Indirect Kinematic Model Transfer

The previous section discussed how to interchange the dependent and independent system parameters if it is difficult to derive the desired model directly. This section will discuss the situation when it is laborious, if not impossible, to arrive at the desired model by direct kinematic model transfer. The procedure involves an intermediate set of generalized coordinates, which are employed to facilitate the determination of the desired model, due to its direct relationship with the desired coordinate set. Hence, the title indirect kinematic model transfer.

Let's assume that it is desired to have the kinematic model of  $\mathcal{Q}$  referenced to a set of generalized coordinates  $\mathcal{Q}$ , but what is available directly is the dependent system parameters  $\mathcal{Q}$  referenced to an initial set of generalized coordinates  $\mathcal{Q}$ . Thus, the initial kinematic model is obtained according to equations 4-1 and 4-2, as

$$\dot{\mathcal{Q}} = [G_{\mathcal{Q}}^{\mathcal{Q}}] \dot{\mathcal{Q}} \quad (4-1)$$

and

$$\ddot{\mathcal{Q}} = [G_{\mathcal{Q}}^{\mathcal{Q}}] \ddot{\mathcal{Q}} + \dot{\mathcal{Q}}^T [H_{\mathcal{Q}\mathcal{Q}}^{\mathcal{Q}}] \dot{\mathcal{Q}} \quad (4-2)$$

Also, it is assumed that the kinematic coefficients of the dependent system parameters  $\mathcal{Q}$  referenced to the initial coordinates  $\mathcal{Q}$  are directly available. This gives

$$\dot{\mathcal{Q}} = [G_{\mathcal{Q}}^{\mathcal{Q}}] \dot{\mathcal{Q}} \quad (4-11)$$

and

$$\ddot{\mathcal{Q}} = [G_{\mathcal{Q}}^{\mathcal{Q}}] \ddot{\mathcal{Q}} + \dot{\mathcal{Q}}^T [H_{\mathcal{Q}\mathcal{Q}}^{\mathcal{Q}}] \dot{\mathcal{Q}} \quad (4-12)$$

However, what is desired is to express the kinematic state of  $\mathcal{Q}$  in terms of  $\mathcal{Q}$  as

$$\dot{\mathbf{q}} = [\mathbf{G}_d^q] \dot{\mathbf{d}} \quad (4-13)$$

and

$$\ddot{\mathbf{q}} = [\mathbf{G}_d^q] \ddot{\mathbf{d}} + \dot{\mathbf{d}}^T [\mathbf{H}_{dd}^q] \dot{\mathbf{d}} \quad (4-14)$$

Rewriting equation 4-1 gives

$$\dot{\Phi} = [\mathbf{G}_\Phi^q]^{-1} \dot{\mathbf{q}} \quad (4-15)$$

Substituting equation 4-15 into equation 4-11 for  $\dot{\Phi}$  gives

$$\dot{\mathbf{d}} = [\mathbf{G}_\Phi^d] [\mathbf{G}_\Phi^q]^{-1} \dot{\mathbf{q}} \quad (4-16)$$

Rearranging equation 4-16 gives

$$\dot{\mathbf{q}} = [\mathbf{G}_\Phi^q] [\mathbf{G}_\Phi^d]^{-1} \dot{\mathbf{d}} \quad (4-17)$$

Comparing equation 4-13 with equation 4-17 yields

$$[\mathbf{G}_d^q] = [\mathbf{G}_\Phi^q] [\mathbf{G}_\Phi^d]^{-1} \quad (4-18)$$

This concludes the indirect transfer of the first-order KIC (ie., the G-functions).

Next, the transfer of the second-order KIC will be derived. Rewriting equation 4-12 as

$$\ddot{\Phi} = [\mathbf{G}_\Phi^d]^{-1} \left( \ddot{\mathbf{d}} - \dot{\Phi}^T [\mathbf{H}_{\Phi\Phi}^d] \dot{\Phi} \right) \quad (4-19)$$

Substituting equation 4-19 into equation 4-2 gives

$$\ddot{\mathbf{q}} = [\mathbf{G}_\Phi^q] [\mathbf{G}_\Phi^d]^{-1} \left( \ddot{\mathbf{d}} - \dot{\Phi}^T [\mathbf{H}_{\Phi\Phi}^d] \dot{\Phi} \right) + \dot{\Phi}^T [\mathbf{H}_{\Phi\Phi}^q] \dot{\Phi} \quad (4-20)$$

Rearranging equation 4-20 and using the generalized dot product gives

$$\ddot{\mathbf{q}} = [\mathbf{G}_{\phi}^q][\mathbf{G}_{\phi}^d]^{-1} \ddot{\mathbf{d}} + \dot{\Phi}^T \left( [\mathbf{H}_{\phi\phi}^q] - [\mathbf{G}_{\phi}^d]^{-1} \cdot [\mathbf{H}_{\phi\phi}^d] \right) \dot{\Phi} \quad (4-21)$$

Substituting equation 4-11 for  $\dot{\Phi}$  into equation 4-21 gives

$$\ddot{\mathbf{q}} = [\mathbf{G}_{\phi}^q][\mathbf{G}_{\phi}^d]^{-1} \ddot{\mathbf{d}} + \dot{\mathbf{d}}^T [\mathbf{G}_{\phi}^d]^{-T} \left( [\mathbf{H}_{\phi\phi}^q] - [\mathbf{G}_{\phi}^d]^{-1} \cdot [\mathbf{H}_{\phi\phi}^d] \right) [\mathbf{G}_{\phi}^d]^{-1} \dot{\mathbf{d}} \quad (4-22)$$

Comparing equation 4-14 with equation 4-22 yields

$$[\mathbf{H}_{dd}^q] = [\mathbf{G}_{\phi}^d]^{-T} \left( [\mathbf{H}_{\phi\phi}^q] - [\mathbf{G}_{\phi}^d]^{-1} \cdot [\mathbf{H}_{\phi\phi}^d] \right) [\mathbf{G}_{\phi}^d]^{-1} \quad (4-23)$$

as the desired second-order KIC. Equation 4-18 and 4-23 give the necessary G- and H-functions for the kinematic model with the dependent system parameter  $\mathbf{q}$  referenced to the desired set of generalized coordinates  $\mathbf{d}$ . This procedure is not limited to the transformation of generalized coordinates involving only one intermediate set of generalized coordinates. There can be as many sets of intermediate generalized coordinates as the situation requires, provided that all the G- and H-functions are interrelated and can readily be determined either directly or indirectly.

#### 4-2 Dynamic Model Transfer

Similar to the kinematic model transfer discussed previously, the readily available dynamic model referenced to an initial set of generalized coordinates given as

$$\mathbf{I}_{\phi} = [\mathbf{I}_{\phi\phi}^{\cdot}] \ddot{\Phi} + \dot{\Phi}^T [\mathbf{P}_{\phi\phi\phi}^{\cdot}] \dot{\Phi} - \mathbf{I}_{\phi}^L \quad (4-24)$$

where  $[\mathbf{I}_{\phi\phi}^{\cdot}]$  and  $[\mathbf{P}_{\phi\phi\phi}^{\cdot}]$  are as defined in equations 3-66 and 3-67, respectively, may not be the desired model. Actually, the dynamic model referenced to another set of generalized coordinates  $\mathbf{d}$  is the ultimate goal. However, it maybe very difficult to

obtain this model directly. Hence, an operation to transfer the dynamic model from the initial set of generalized coordinates to the desired coordinates set is necessary.

It is assumed that the kinematic influence coefficients between the initial and the desired coordinates sets are available in the following form

$$\dot{\underline{d}} = [G_{\varphi}^d] \dot{\underline{\varphi}} \quad (4-25)$$

and

$$\ddot{\underline{d}} = [G_{\varphi}^d] \ddot{\underline{\varphi}} + \dot{\underline{\varphi}}^T [H_{\varphi\varphi}^d] \dot{\underline{\varphi}} \quad (4-26)$$

Using the principle of virtual work, the dynamic equation can be expressed in terms of the desired input set  $\underline{d}$  as

$$\underline{I}_d = [G_{\varphi}^d]^{-T} \underline{I}_{\varphi} \quad (4-27)$$

or

$$\underline{I}_d = [G_{\varphi}^d]^{-T} \left( [I_{\varphi\varphi}^{\dot{}}] \ddot{\underline{\varphi}} + \dot{\underline{\varphi}}^T [P_{\varphi\varphi\varphi}^{\dot{}}] \dot{\underline{\varphi}} - \underline{I}_{\varphi}^L \right) \quad (4-28)$$

Substituting equation 4-6, with  $q$  replaced by  $d$ , for  $\ddot{\underline{\varphi}}$  into equation 4-28, rearranging, and using the generalized dot product gives

$$\begin{aligned} \underline{I}_d &= [G_{\varphi}^d]^{-T} [I_{\varphi\varphi}^{\dot{}}] [G_{\varphi}^d]^{-1} \ddot{\underline{d}} \\ &+ \dot{\underline{\varphi}}^T \left\{ [G_{\varphi}^d]^{-T} \cdot [P_{\varphi\varphi\varphi}^{\dot{}}] - \left( [G_{\varphi}^d]^{-T} [I_{\varphi\varphi}^{\dot{}}] [G_{\varphi}^d]^{-1} \right) \cdot [H_{\varphi\varphi}^d] \right\} \dot{\underline{d}} \\ &- [G_{\varphi}^d]^{-T} \underline{I}_{\varphi}^L \end{aligned} \quad (4-29)$$

Substituting equation 4-5, with  $q$  replaced by  $d$ , for  $\dot{\underline{\varphi}}$  and  $\dot{\underline{\varphi}}^T$  into equation 4-29 yields

$$\begin{aligned}
\mathbf{T}_d = & \left[ \mathbf{G}_\phi^d \right]^{-T} \left[ \dot{\mathbf{I}}_{\phi\phi} \right] \left[ \mathbf{G}_\phi^d \right]^{-1} \ddot{\mathbf{d}} \\
& + \ddot{\mathbf{d}} \left[ \mathbf{G}_\phi^d \right]^{-T} \left\{ \left( \left[ \mathbf{G}_\phi^d \right]^{-T} \cdot \left[ \dot{\mathbf{P}}_{\phi\phi\phi} \right] \right) \right. \\
& \left. - \left( \left[ \mathbf{G}_\phi^d \right]^{-T} \left[ \dot{\mathbf{I}}_{\phi\phi} \right] \left[ \mathbf{G}_\phi^d \right]^{-1} \right) \cdot \left[ \mathbf{H}_{\phi\phi}^d \right] \right\} \left[ \mathbf{G}_\phi^d \right]^{-1} \dot{\mathbf{d}} - \left[ \mathbf{G}_\phi^d \right]^{-T} \mathbf{T}_\phi^L
\end{aligned} \quad (4-30)$$

Comparing the form of equation 4-30 with that of equation 4-24, the dynamic equation referenced to the desired set of generalized coordinates ( $\mathbf{d}$ ) can be expressed as

$$\mathbf{T}_d = \left[ \dot{\mathbf{I}}_{dd} \right] \ddot{\mathbf{d}} + \dot{\mathbf{d}} \left[ \dot{\mathbf{P}}_{ddd} \right] \dot{\mathbf{d}} - \mathbf{T}_d^L \quad (4-31)$$

where

$$\left[ \dot{\mathbf{I}}_{dd} \right] = \left[ \mathbf{G}_\phi^d \right]^{-T} \left[ \dot{\mathbf{I}}_{\phi\phi} \right] \left[ \mathbf{G}_\phi^d \right]^{-1} \quad (4-32)$$

$$\left[ \dot{\mathbf{P}}_{ddd} \right] = \left[ \mathbf{G}_\phi^d \right]^{-T} \left\{ \left[ \mathbf{G}_\phi^d \right]^{-T} \cdot \left[ \dot{\mathbf{P}}_{\phi\phi\phi} \right] - \left[ \dot{\mathbf{I}}_{dd} \right] \cdot \left[ \mathbf{H}_{\phi\phi}^d \right] \right\} \left[ \mathbf{G}_\phi^d \right]^{-1} \quad (4-33)$$

and

$$\mathbf{T}_d^L = \left[ \mathbf{G}_\phi^d \right]^{-T} \mathbf{T}_\phi^L \quad (4-34)$$

Note that utilizing this generalized transfer, any readily obtained dynamic model can be transferred to any desired set of generalized coordinates once the KIC's relating  $\mathbf{d}$  to  $\phi$  are obtained.

For the purpose of discussion, the above mentioned procedure is not the only way to arrive at the model referenced to the desired set of generalized coordinates. An alternate procedure involves the derivation of intermediate dynamic models. Instead of finding the kinematic model relating the initial set to the desired set of generalized coordinates, then performing the dynamic model transfer as before, the dynamic model at each stage is determined. This means that if  $\left[ \mathbf{G}_\phi^q \right]$ ,  $\left[ \mathbf{H}_{\phi\phi}^q \right]$ ,  $\left[ \mathbf{G}_d^q \right]$ , and  $\left[ \mathbf{H}_{dd}^q \right]$  are known, where  $\phi$ ,  $q$ , and  $d$  are as defined before,  $\mathbf{T}_d$  is found in terms of  $\mathbf{T}_q$  which is in turn found in terms of  $\mathbf{T}_\phi$ . First, it is assumed that

$\mathbf{I}_\phi$  is known. Second, transfer the dynamic model to the intermediate set of generalized coordinates as

$$\mathbf{I}_q = [\mathbf{G}_\phi^q]^{-T} \mathbf{I}_\phi \quad (4-35)$$

Finally, transferring from the intermediate set to the desired set of generalized coordinates yields

$$\mathbf{I}_d = [\mathbf{G}_q^d]^{-T} \mathbf{I}_q \quad (4-36)$$

as the desired dynamic model. In equations 4-35 and 4-36, procedures similar to those of equations 4-24 through 4-34 are implied.

Both the procedures are valid as shown in Freeman and Tesar[14]. The decision to choose either one of them is entirely up to the users. It depends on what variable relations are available and/or which is easier to obtain. However, the first approach is suggested by Freeman and Tesar[14] and Hudgens [21] for design and control purposes since it maybe necessary to investigate more than one set of generalized coordinates.

#### 4-3 Application of Transfer of Generalized Coordinates to Multi-loop Parallel Mechanisms

Generally, the load capacity of serial manipulators is limited by the size of the actuators (Driga, Eppes and Flake [10]). Hence, for high load capacity, hydraulic actuators are commonly used due to their high load to weight ratio. However, the main shortcoming for using hydraulic actuators is low precision. As mentioned before, the error in serial manipulators is additive. If hydraulic actuators are used in serial manipulators, the inaccuracy will be even larger. Therefore, for any robot with high load capacity and acceptable precision, parallel structure robots are frequently used.

Chapter 2 gave a brief discussion of the work on parallel mechanisms that has been done by other researchers. This section will introduce a different

approach to parallel mechanism modeling using the method of KIC along with the transfer of generalized coordinates. However, unlike the open-loop serial manipulators that have been discussed so far, parallel manipulator structures are not as straight forward as the serial manipulators due to the fact that there is more than one way to form a parallel manipulator. The discussion here will concentrate on mechanisms that can be classified as "fully parallel," such as the generalized Stewart Platform shown in Fig. 4-1. Although the figure shows six legs connected to a common platform with each of the legs having six DOF, the procedure which will be discussed shortly is equally applicable to mechanisms having any number of legs, provided that each of the link has the same number of DOF as does the operational space.

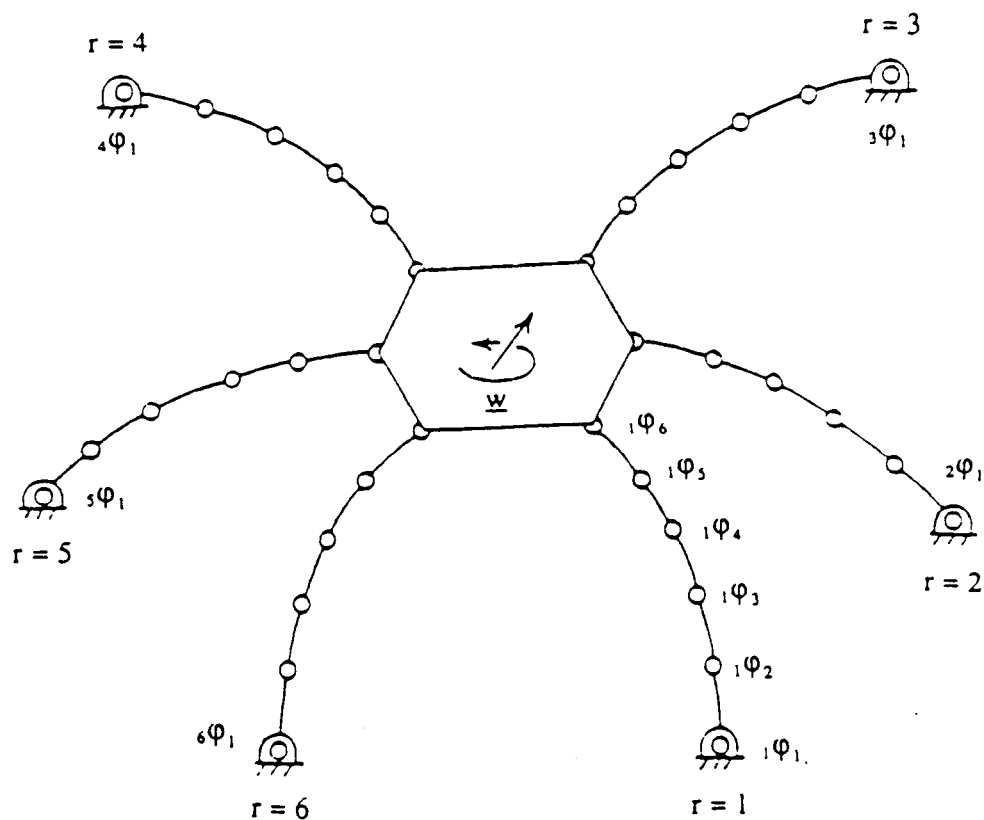
The generalized Stewart Platform in Fig. 4-1 has six DOF at the platform, but it has thirty-six joints. To control this mechanism, one has to decide which of the six out of the thirty-six potential input locations to choose. For a start, examining the mechanism a little closer shows that it actually consists of six serial mechanisms connected to a common base and a common platform. Similar to the procedure discussed previously for serial manipulators, the first step is to obtain a model directly for each leg with respect to a set of generalized coordinates that can easily be determined. By treating the other five legs as if they do not exist, each leg is modeled with respect to its own joint coordinate set  ${}^r\mathcal{Q}$ , where  $r(r=1, 2, \dots, 6)$  is the leg number. This yields results similar to the serial manipulator discussed in the previous sections, ie.,

$$\left[ {}^rG_{\mathcal{Q}}^q \right], \left[ {}^rH_{\mathcal{Q}\mathcal{Q}}^q \right], \left[ {}^rI_{\mathcal{Q}\mathcal{Q}}^{\dot{}} \right], \left[ {}^rP_{\mathcal{Q}\mathcal{Q}\mathcal{Q}}^{\dot{}} \right], \left[ {}^rT_{\mathcal{Q}}^L \right]$$

where  $\mathcal{Q}$  represents the intermediate set of generalized coordinates associated with the six parameters describing the motion of the platform. Next, apply the direct kinematic transfer of equations 4-9 and 4-10 to give

$$\left[ {}^rG_{\mathcal{Q}}^p \right] = \left[ {}^rG_{\mathcal{Q}}^q \right]^{-1} \quad (4-37)$$

and



*Desired* Generalized Coordinates

$$\underline{d} = \left[ {}_1\varphi_1, {}_2\varphi_1, {}_3\varphi_1, {}_4\varphi_1, {}_5\varphi_1, {}_6\varphi_1 \right]^T$$

*Initial* Generalized Coordinates

$${}_r\varphi = \left[ {}_r\varphi_1, {}_r\varphi_2, {}_r\varphi_3, {}_r\varphi_4, {}_r\varphi_5, {}_r\varphi_6 \right]^T ; \quad r = 1, 2, \dots, 6$$

**Fig. 4-1 Generalized Stewart Platform\***

\*Adapted from Freeman and Tesar [14]

$$\begin{bmatrix} {}^r H_{qq}^{\phi} \end{bmatrix} = - \begin{bmatrix} {}^r G_{\phi}^q \end{bmatrix}^T \left( \begin{bmatrix} {}^r G_{\phi}^q \end{bmatrix}^{-1} \bullet \begin{bmatrix} {}^r H_{\phi\phi}^q \end{bmatrix} \right) \begin{bmatrix} {}^r G_{\phi}^q \end{bmatrix}^{-1} \quad (4-38)$$

From the transfer of the dynamic model as in Section 4-2, one has

$$\begin{bmatrix} {}^r \dot{I}_{qq} \end{bmatrix} = \begin{bmatrix} {}^r G_{\phi}^q \end{bmatrix}^T \begin{bmatrix} {}^r \dot{I}_{\phi\phi} \end{bmatrix} \begin{bmatrix} {}^r G_{\phi}^q \end{bmatrix}^{-1} \quad (4-39)$$

$$\begin{aligned} \begin{bmatrix} {}^r P_{qqq} \end{bmatrix} &= \begin{bmatrix} {}^r G_{\phi}^q \end{bmatrix}^T \left\{ \left( \begin{bmatrix} {}^r G_{\phi}^q \end{bmatrix}^{-T} \bullet \begin{bmatrix} {}^r P_{\phi\phi\phi} \end{bmatrix} \right) \right. \\ &\quad \left. - \left( \begin{bmatrix} {}^r \dot{I}_{qq} \end{bmatrix} \bullet \begin{bmatrix} {}^r H_{\phi\phi}^q \end{bmatrix} \right) \right\} \begin{bmatrix} {}^r G_{\phi}^q \end{bmatrix}^{-1} \end{aligned} \quad (4-40)$$

and

$${}^r \mathbf{I}_q^L = \begin{bmatrix} {}^r G_{\phi}^q \end{bmatrix}^{-T} {}^r \mathbf{I}_{\phi}^L \quad (4-41)$$

Notice that until this point, nothing has been said about the platform itself. Actually, the model of the platform can be included in one of the legs in the above derivation. However, it is felt that separating the platform model from the legs gives a more explicit physical meaning in the final formulation. Now that all the legs are referenced to the same intermediate set of generalized coordinates, they can be combined together along with the inertia effect of the platform and any load applied to the platform. As a result, the dynamic model can be expressed as

$$\begin{bmatrix} \dot{I}_{qq} \end{bmatrix} = \begin{bmatrix} I_{qq} \end{bmatrix} + \sum_{r=1}^6 \begin{bmatrix} {}^r \dot{I}_{qq} \end{bmatrix} \quad (4-42)$$

$$\begin{bmatrix} P_{qqq} \end{bmatrix} = \begin{bmatrix} P_{qqq} \end{bmatrix} + \sum_{r=1}^6 \begin{bmatrix} {}^r P_{qqq} \end{bmatrix} \quad (4-43)$$

and

$$\begin{aligned}
{}^r\mathbf{I}_q^L &= [{}^r\mathbf{G}_\phi^q]^{-T} {}^r\mathbf{I}_\phi^L \\
&= \sum_{r=1}^6 {}^r\mathbf{I}^L + \left\{ \frac{{}^6\mathbf{f}^c}{\mathbf{m}^{67}} \right\}
\end{aligned} \tag{4-44}$$

where  ${}^6\mathbf{f}^c$  and  $\mathbf{m}^{67}$  are a set of external forces and moments applied to the platform, respectively. The terms,  $[\mathbf{I}_{qq}]$  and  $[\mathbf{P}_{qqq}]$  are the platform effective inertia matrix and inertia power array, respectively, defined as

$$[\mathbf{I}_{qq}] = \begin{bmatrix} \mathbf{M}^{67} & 0 & 0 & \mathbf{Q}^T \\ 0 & \mathbf{M}^{67} & 0 & \mathbf{Q}^T \\ 0 & 0 & \mathbf{M}^{67} & \mathbf{Q}^T \\ \mathbf{Q} & \mathbf{Q} & \mathbf{Q} & [\Pi^{67}] \end{bmatrix} \tag{4-45}$$

$$[\mathbf{P}_{qqq}]_{k;;} = \begin{bmatrix} [0] & [0] \\ [0] & [\mathbf{P}_{kqq}] \end{bmatrix}, \quad k = 4, 5, 6 \tag{4-46}$$

where

$$[\Pi^{67}] = \left[ \begin{array}{c|c|c} [\Pi^{67}]_{;1} & [\Pi^{67}]_{;2} & [\Pi^{67}]_{;3} \end{array} \right] \tag{4-47}$$

and the first three planes of  $[\mathbf{P}_{qqq}]$  are 6 by 6 null matrices, and

$$[\mathbf{P}_{4qq}] = \left[ \begin{array}{c|c|c} \mathbf{Q} & [\Pi^{67}]_{3;}^T & -[\Pi^{67}]_{2;}^T \end{array} \right] \tag{4-48}$$

$$[\mathbf{P}_{5qq}] = \left[ \begin{array}{c|c|c} -[\Pi^{67}]_{3;}^T & \mathbf{Q} & [\Pi^{67}]_{1;}^T \end{array} \right] \tag{4-49}$$

$$[P_{6qq}] = \left[ \begin{array}{c|c|c} [\Pi^{67}]_{2;}^T & -[\Pi^{67}]_{1;}^T & Q \end{array} \right] \quad (4-50)$$

are the components in the last three planes. Equations 4-48 to 50 are all 3 by 3 matrices.

Up to this point the kinematic and dynamic model of the generalized Stewart Platform has been formed referenced to the common set of generalized coordinates  $Q$ . This common reference gives the flexibility to transfer the model to any six of the thirty-six potential joint inputs. One may desire to have a single leg provide control of all the six DOF or one may desire to control only one DOF from each leg. For the sole purpose of this discussion, let's assume that it is desired to have the first joint of leg one, the second joint of leg two, and so on, provide control of the six DOF. This gives the desired generalized input set as

$$\underline{d} = \left[ {}_1\psi_1, {}_2\psi_2, \dots, {}_6\psi_6 \right]^T \quad (4-51)$$

To obtain the first- and second-order KIC relating the desired set of generalized coordinates  $\underline{d}$  to the intermediate set  $Q$ , simply extract the corresponding row of the G-functions and plane of the H-functions, from equations 4-37 and 4-38 for the respective legs. This extraction yields

$$[G_q^d] = \left[ \begin{array}{c} \left[ {}^1G_q^\psi \right]_{1;} \\ \left[ {}^2G_q^\psi \right]_{2;} \\ \left[ {}^3G_q^\psi \right]_{3;} \\ \left[ {}^4G_q^\psi \right]_{4;} \\ \left[ {}^5G_q^\psi \right]_{5;} \\ \left[ {}^6G_q^\psi \right]_{6;} \end{array} \right] \quad (4-52)$$

and

$$[H_{qq}^d] = \begin{bmatrix} [{}^1H_{qq}^\varphi]_{1;;} \\ [{}^2H_{qq}^\varphi]_{2;;} \\ [{}^3H_{qq}^\varphi]_{3;;} \\ [{}^4H_{qq}^\varphi]_{4;;} \\ [{}^5H_{qq}^\varphi]_{5;;} \\ [{}^6H_{qq}^\varphi]_{6;;} \end{bmatrix} \quad (4-53)$$

Having formed the KIC required by the transfer equations, the desired dynamic equation can be written as

$$\mathbb{I}_d = [\dot{I}_{dd}] \ddot{d} + \dot{d}^T [\dot{P}_{ddd}] \dot{d} - \mathbb{I}_d^L \quad (4-54)$$

where

$$[\dot{I}_{dd}] = [G_q^d]^T [\dot{I}_{qq}] [G_q^d]^{-1} \quad (4-55)$$

$$[\dot{P}_{ddd}] = [G_q^d]^T \left\{ \left( [G_q^d]^{-T} \cdot [\dot{P}_{qqq}] \right) - \left( [\dot{I}_{dd}] \cdot [H_{qq}^d] \right) \right\} [G_q^d]^{-1} \quad (4-56)$$

and

$$\mathbb{I}_d^L = [G_q^d]^{-T} \mathbb{I}_q^L \quad (4-57)$$

The above derivation shows how to model multi-loop parallel mechanisms in terms of the relatively simple serial manipulator model. The beauty of this procedure is that serial and parallel mechanisms can be treated in exactly the same way. All that is needed are the isomorphic transformation equations. Throughout the derivation, the desired equations are always in the same basic form, i.e., time dependent terms are separated from the configuration dependent terms.

The G- and H-functions along with the  $[I^\bullet]$  matrix and  $[P^\bullet]$  array also maintain the same form throughout, hence the term isomorphic.

The procedure discussed in this section for multi-loop fully parallel mechanisms can be summarized in the following steps :

- (a) number all the legs and joint inputs;
- (b) determine the kinematic and dynamic model of every leg, in terms of the easily obtained set of joint coordinates, by treating each of them as a single serial linkage;
- (c) perform the generalized coordinate transformation on each leg from joint coordinates to a common set of generalized coordinates;
- (d) include the inertia effect of the platform and any applied load;
- (e) prepare to transfer the model to the desired reference by extracting the respective row from the G-functions, and plane from the H-functions;
- (f) form the desired dynamic equation by applying the transfer equations.

In terms of symbols, the above procedure can be summarized and represented as

$$[{}_r G_\phi^q], [{}_r H_{\phi\phi}^q], [{}_r I_{\phi\phi}^\bullet], [{}_r P_{\phi\phi\phi}^\bullet], {}_r T_\phi^L$$

↓

$$[{}_r G_q^\phi], [{}_r H_{qq}^\phi], [{}_r I_{qq}^\bullet], [{}_r P_{qqq}^\bullet], {}_r T_q^L$$

↓

$$[I_{qq}^\bullet], [P_{qqq}^\bullet]$$

↓

$$[G_d^q], [H_{dd}^q]$$

$$\downarrow$$

$$[\dot{I}_{dd}], [\dot{P}_{ddd}], T_d^L$$

This concludes the development of the kinematic and dynamic models of the generalized Stewart Platform using the method of KIC together with the isomorphic transformation of generalized coordinates. In the next chapter, all the techniques and procedures discussed so far will be used to model the Dynamic Docking Test System, which was described briefly in Chapter 2.

## CHAPTER 5

### KINEMATIC AND DYNAMIC MODELING OF THE DYNAMIC DOCKING TEST SYSTEM

As discussed in Section 2-2, the Dynamic Docking Test System(DDTS) is a mechanism consisting of a base and a platform connected by six legs as shown in Fig. 2-5. All the legs are made up of prismatic joints with one end of the leg connected to the platform by a ball joint and the other end connected to the base by two intersecting revolute joints(or hooke joint). It is obvious that the structural arrangement of the DDTS is a variation of the generalized Stewart Platform addressed in Section 4-3.

A conceptual view of the DDTS showing the relative position of all the connection points for the legs on the platform as well as the base is given in Fig. 5-1. Notice that each leg has six DOF, three from the ball joint, two from the hooke joint and one from the prismatic joint. Although there can be one actuator for each DOF, the physical structure of the mechanism is designed such that only one actuator, located at the prismatic joints, is available for each leg. Therefore, the ball and hooke joints are free to rotate about their own axes. The following sections will discuss the development of the model, starting from coordinate frame definition to the final dynamic model.

#### 5-1 Kinematic Model

The approach taken here is different from that of Hudgens [21]. The main difference is that they considered the three generalized coordinates for the ball joint, with the X-axis of the last joint pointing towards the center of the platform, the two generalized coordinates for the hooke joint, and the global reference coordinate(or the base reference coordinate) together as one complete set of generalized coordinates. These six coordinates make up their initial set. On the other hand, the initial coordinates adopted in this work consist of three generalized coordinates, two located at the hooke joint and one at prismatic link, as shown in

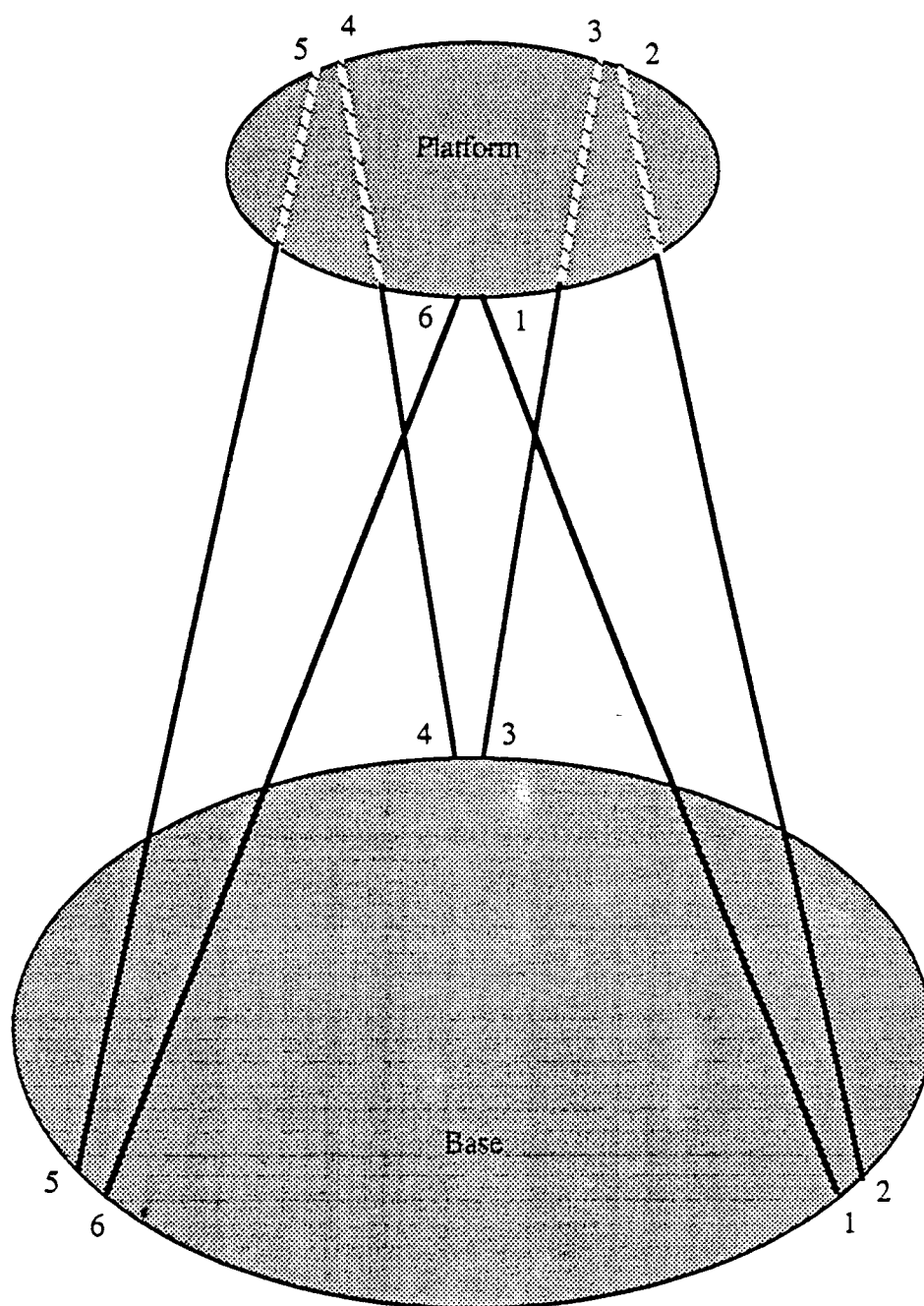


Fig. 5-1 Legs Arrangement

Fig. 5-2. This requires a slightly different set of transfer equations to determine the platform based intermediate model, which will be discussed in Section 5-3.

It is felt that separating the coordinates into small subsets will facilitate the derivation and subsequently the computation. As a matter of fact, in the initial model, instead of 6 by 6 matrices, only 3 by 3 matrices need to be dealt with. This is made possible by neglecting the three DOF of the ball joint, which will constitute three mutually orthogonal rotational axes. To neglect the ball joint in the first set of coordinates, the first-order KIC relating the hooke and prismatic joints to the platform must be independent of the three pseudo axes of the ball joint. This independency can be shown by employing the matrix partitioning technique to the G-functions as illustrated in Freeman [16]. However, the one shortcoming that arose from dividing into three sets of generalized coordinates is that it was necessary to perform several of the previously discussed generalized coordinate transfers. Since all the six legs are identical, only one of them need be discussed in the development of the total system model.

#### 5-1.1 System Definition

As mentioned before, there are three DOF for each leg(neglecting the ball joint). Two of the freedoms are from the hooke joint, and the other one is from the prismatic joint. Following the convention adopted in Chapter 3, Section 3-2.1, the three coordinate frames for each leg are set up as shown in Fig. 5-2. The fixed frame for each leg is indicated as X, Y and Z in the figure. This frame is fixed relative to each leg so that it behaves like a local global frame for each individual leg. To transform the expression to the globally referenced frame at the center of the base, one needs only to multiply the local global frame results by a rotation matrix.

Before all the different reference frames get too complicated, it is appropriate to define each of the terms that will be used throughout this work. From now on, leg frame refers to the fixed frame for each leg(or the local leg frame in the above paragraph), which is represented by the X, Y and Z orthogonal unit vectors. Base frame is referred to as the world coordinate frame and is located at the center of the base.  $X^*$ ,  $Y^*$  and  $Z^*$  are used to represent the three mutually orthogonal unit

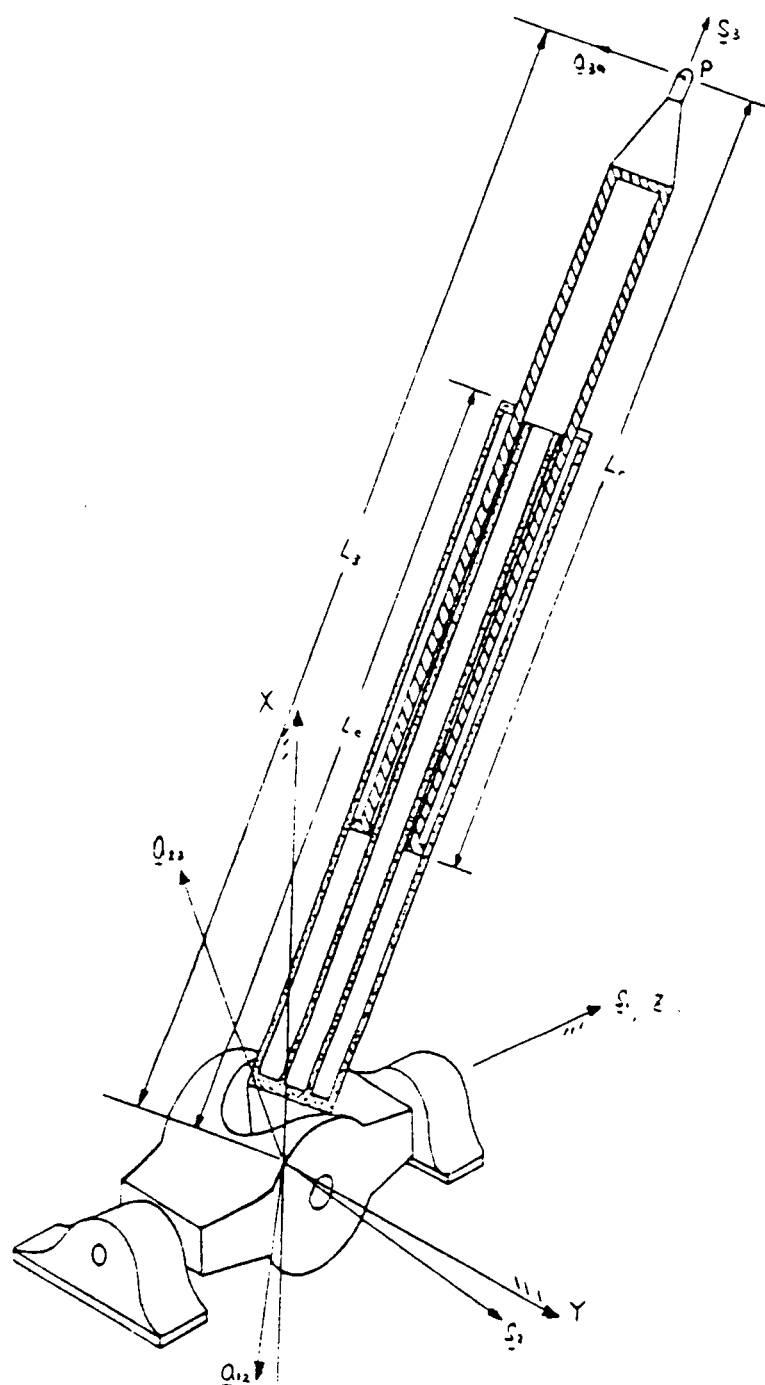


Fig. 5-2 Kinematic Representation of One of the Legs of the Dynamic Docking Test System

vectors for the base frame. Also, platform frame refers to the coordinate frame located at the center of the platform, represented by the lower case x, y and z, which will be discussed in a later section. The three coordinate frames mentioned above are shown in Fig. 5-3.

Having a leg frame for each individual leg facilitates the modeling procedure. Since, with respect to this frame, all the six legs can be modeled identically. Referring back to Fig. 5-2,  $L_c$  is defined as the length of the piston cylinder,  $L_r$  is the length of the piston rod, whereas,  $L_3$  is the total length of the piston, measured from the origin of the leg frame to the end of the prismatic link. Notice that  $L_c$  and  $L_r$  are fixed length but  $L_3$  is variable.

Following the set up in Fig. 5-2 and the convention in Section 3-2.1, a simplified version of Fig. 5-2 can be made as illustrated in Fig. 5-4 along with the resulting link parameters. Defining

$$\underline{s}_j = (X_j, Y_j, Z_j) ; j = 1, 2, 3 \quad (5-1)$$

and premultiplying the locally referenced vector  ${}^{(j)}\underline{s}_j$  by the rotation matrix T,

$$T_j = \begin{bmatrix} X_{(j-1)j} & Y_{(j-1)j}Z_{(j-1)j} - Z_{j-1}Y_{(j-1)j} & X_{j-1} \\ Y_{(j-1)j} & Z_{(j-1)j}X_{(j-1)j} - X_{j-1}Z_{(j-1)j} & Y_{j-1} \\ Z_{(j-1)j} & X_{(j-1)j}Y_{(j-1)j} - Y_{j-1}X_{(j-1)j} & Z_{j-1} \end{bmatrix} \quad (5-2)$$

gives

$$\underline{s}_1 = \begin{bmatrix} 0 \\ 0 \\ 1 \end{bmatrix}, \quad \underline{s}_2 = \begin{bmatrix} S_{\theta_1} \\ -C_{\theta_1} \\ 0 \end{bmatrix}, \quad \underline{s}_3 = \begin{bmatrix} C_{\theta_1}S_{\theta_2} \\ S_{\theta_1}S_{\theta_2} \\ -C_{\theta_2} \end{bmatrix} \quad (5-3)$$

where

$$S_{\theta_i} = \sin(\theta_i), \quad C_{\theta_i} = \cos(\theta_i)$$

Similarly, defining

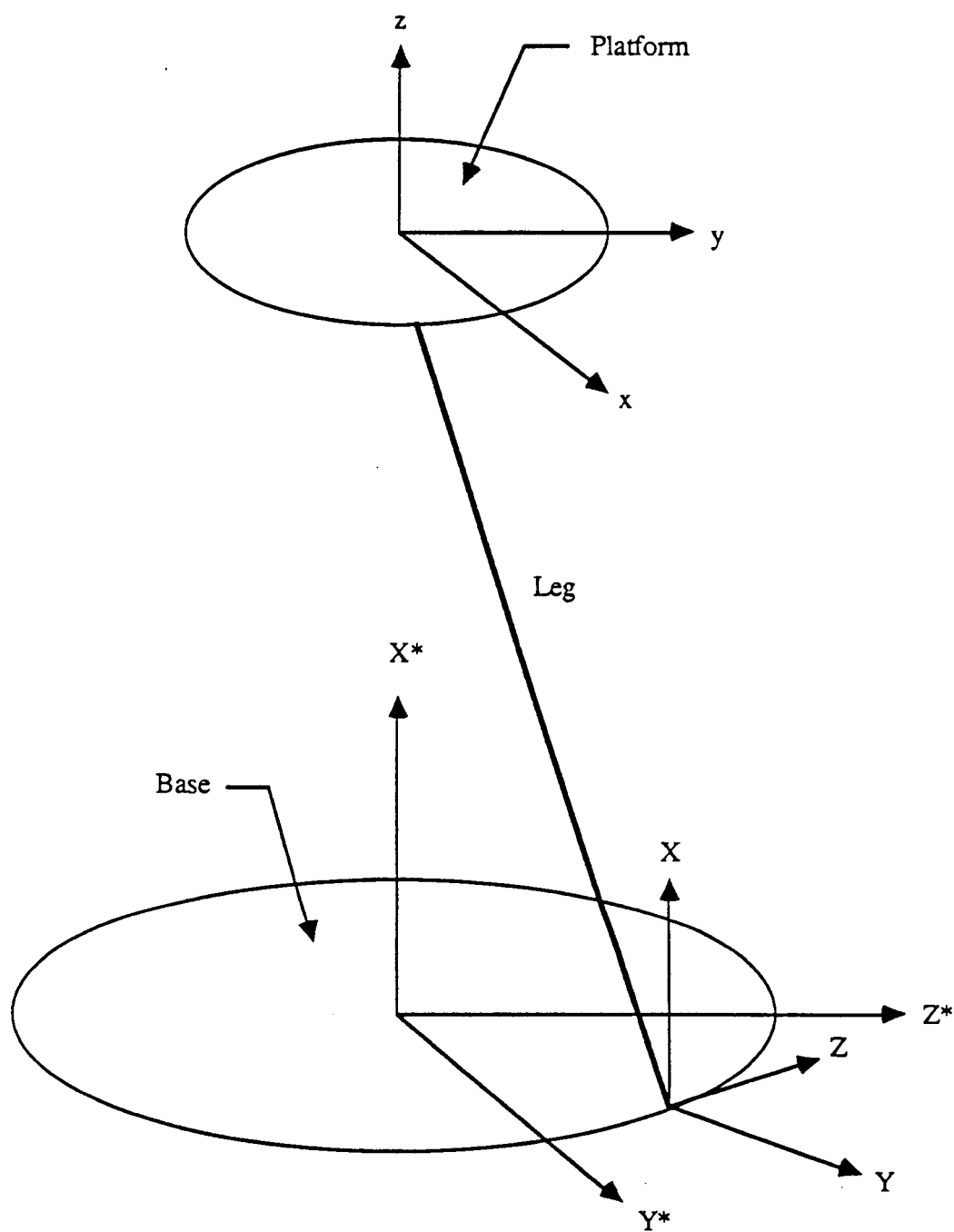
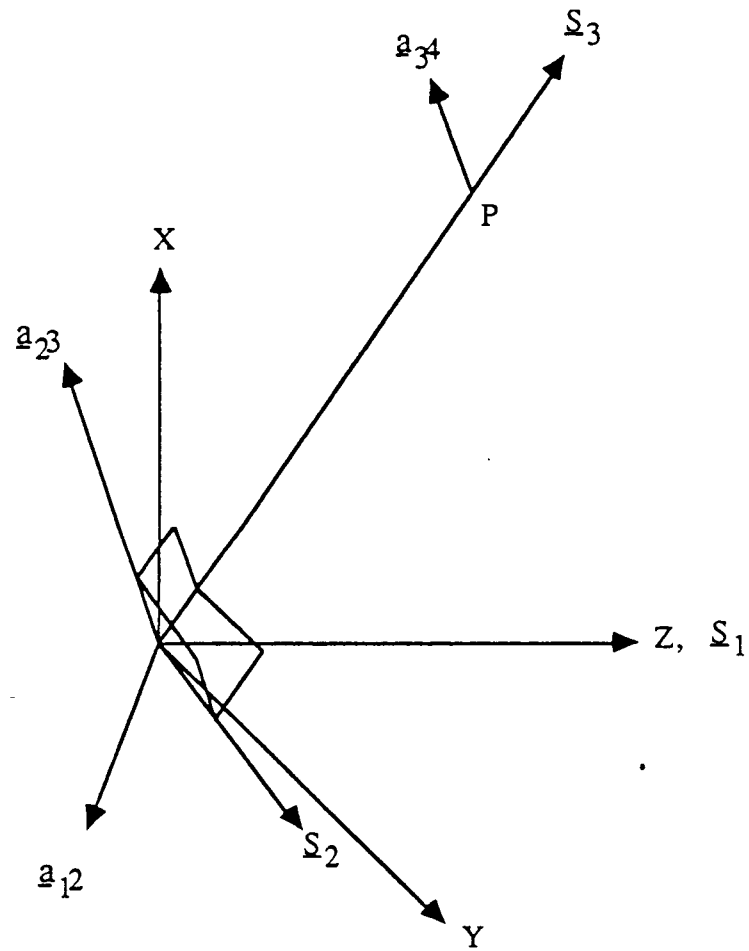


Fig. 5-3 Coordinate Frames Used for the DDTS



$j$	$S_{jj}$	$a_{jk}$	$\alpha_{jk}$	$\theta_j$
1	0	0	$90^\circ$	$\theta_1$
2	0	0	$90^\circ$	$\theta_2$
3	$L_3$	0	~	0

Fig. 5-4 Link Parameters for Each Leg

$$\mathbf{a}_{jk} = (X_{jk}, Y_{jk}, Z_{jk}) ; j = 1, 2, 3 ; k = j + 1 \quad (5-4)$$

and premultiplying its local vector by the matrix T in equation 5-2 gives

$$\mathbf{a}_{12} = \begin{bmatrix} C_{\theta_1} \\ -S_{\theta_1} \\ 0 \end{bmatrix}, \quad \mathbf{a}_{23} = \begin{bmatrix} C_{\theta_1} C_{\theta_2} \\ S_{\theta_1} C_{\theta_2} \\ S_{\theta_2} \end{bmatrix}, \quad \mathbf{a}_{34} = \begin{bmatrix} C_{\theta_1} C_{\theta_2} \\ S_{\theta_1} C_{\theta_2} \\ S_{\theta_2} \end{bmatrix} \quad (5-5)$$

Notice that  $\mathbf{a}_{23}$  and  $\mathbf{a}_{34}$  are identical. This is expected since  $\mathbf{a}_{34}$  is in the last link and can be arbitrarily chosen to be parallel to  $\mathbf{a}_{23}$ .

### 5-1.2 Specification of the Platform Position and Orientation

To begin the kinematic analysis of the mechanism, it is assumed that the kinematic state of the end-effector(or the platform) is known. For any rigid body in space, its position can easily be specified by a vector pointing from the origin of the reference frame to a fixed point in the body. However, there are many ways to specify the orientation of the rigid body as discussed in Hudgens [21]. Euler angles, Cayley-Klein parameters and quaternions are some of the possible ways.

The method employed in this work to specify the position of the platform is by specifying the vector pointing from the origin of the base frame to the center of the platform. The orientation of the platform is given by a pair of orthogonal unit vectors fixed at the center of the platform. One of these vectors is normal to the surface of the platform. The vector cross product of these two unit vectors gives the third vector to make up the three mutually orthogonal unit vectors, which define the platform frame. Fig. 5-5 illustrates the specification for the position and orientation of the platform. Vectors  $\mathbf{a}$  and  $\mathbf{s}$  represent the two mutually orthogonal unit vectors used to specify the location of x and z axes, respectively, of the platform frame in the base frame.

Once the position and orientation of the platform is known, the positions of all the six connecting points (ie., the ball joints), which are fixed with

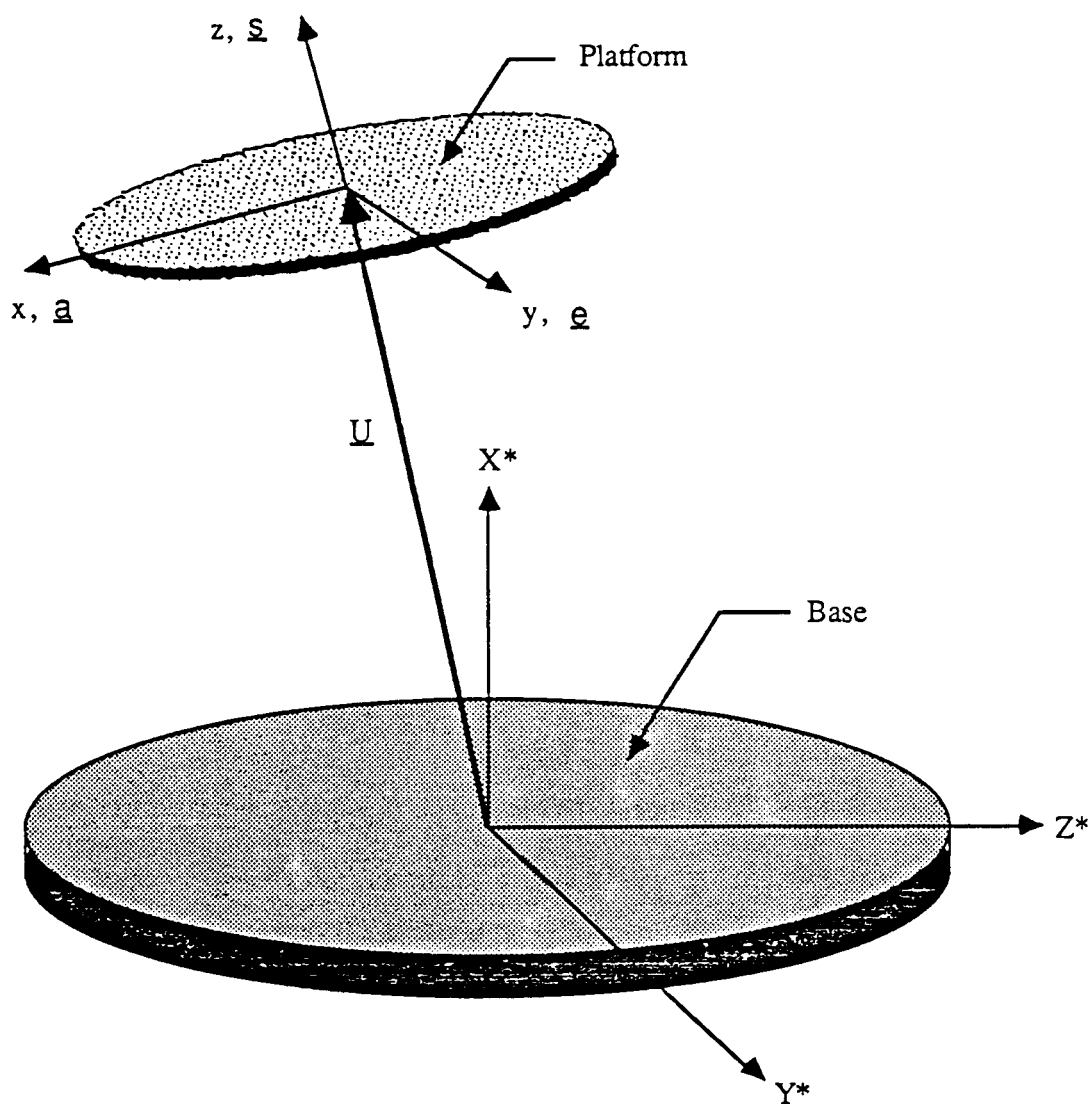


Fig. 5-5 Specification of the Platform Position and Orientation

respect to the platform frame, can now be transformed to the base frame by a simple rotation matrix  $T$ ,

$$T = \begin{bmatrix} \underline{a} & \underline{s} \times \underline{a} & \underline{s} \end{bmatrix} \quad (5-6)$$

and a linear translation vector

$$\underline{U} = [U_x \cdot \ U_y \cdot \ U_z \cdot]^T \quad (5-7)$$

Note that equation 5-6 is in the basic form of equation 5-2.

### 5-1.3 Reverse Kinematics of Each Leg

The coordinate frames for each leg are set up in the previous section. Furthermore, vectors  $\underline{S}_i$  and  $\underline{A}_{ik}$  for each joint are determined using equations 5-3 and 5-5. However, as stated in Chapter 3 that in all practical purposes, the end-effector (or in this case the point P in Fig. 5-1) kinematic state is usually known instead of the state at each joint. Thus, this section will discuss the reverse kinematic (position only) of each link via the geometric approach.

The geometry of each leg can be represented as in Fig. 5-6. The origin of the leg is always at the origin of the leg frame. The other end of the leg is located at point P ( $X_p$ ,  $Y_p$  and  $Z_p$ ), the ball joint, which is determined by knowing the specified platform position and orientation and the location of the point P relative to the platform frame. As indicated in the figure, each leg can be characterized by two parameters (ie.,  $\Psi_1$  and  $\Psi_2$ ). The length of the leg can be written as

$$L_3 = \sqrt{X_p^2 + Y_p^2 + Z_p^2} \quad (5-8)$$

Also,

$$\Psi_1 = \cos^{-1} \left( \frac{Z_p}{L_3} \right) \quad (5-9)$$

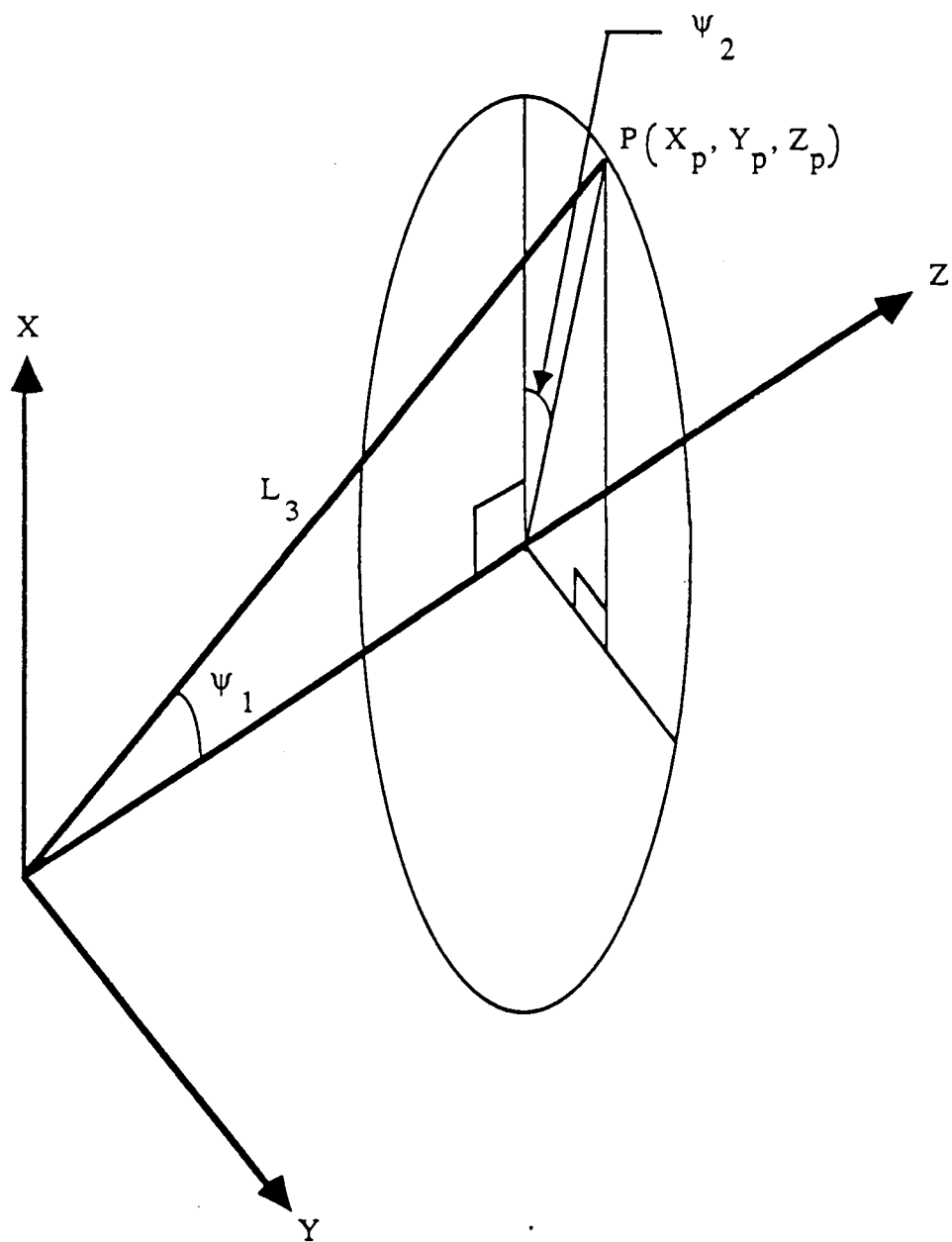


Fig. 5-6 Reverse Position Kinematic of the Leg

and

$$\psi_2 = \sin^{-1} \left( \frac{Y_p}{L_3 \sin \psi_1} \right) \quad (5-10)$$

By geometry and the definition of  $\theta_1$  and  $\theta_2$ , it can be shown that

$$\theta_1 = 180 + \psi_2 \quad (5-11)$$

and

$$\theta_2 = 180 + \psi_1 \quad (5-12)$$

Equations 5-11 and 5-12 provide the solutions for the unknowns in equations 5-3 and 5-5.

#### 5-1.4 Initial First- and Second-order Kinematics of Each Leg

From the given conditions, treating the relative joint parameters as the independent system parameters, all necessary G- and H-functions can be easily obtained directly. Recalling equations 3-24 to 3-34, the velocity and acceleration of the ball joint can be written as

$${}_r\dot{\mathbf{P}} = \left[ {}_r\mathbf{G}_\Phi^p \right] {}_r\dot{\Phi} \quad (5-13)$$

$${}_r\ddot{\mathbf{P}} = \left[ {}_r\mathbf{G}_\Phi^p \right] {}_r\ddot{\Phi} + {}_r\dot{\Phi}^T \left[ {}_r\mathbf{H}_{\Phi\Phi}^p \right] {}_r\dot{\Phi} \quad (5-14)$$

and

$${}_r\Phi = \left( {}_r\theta_1, {}_r\theta_2, {}_rL_3 \right) \quad (5-15)$$

where  $r(r=1,2,\dots,6)$  is the leg number. However, instead of the 6 by 6 matrix for the G-functions and 6 by 6 by 6 array for the H-functions as before, the G-function here is a 3 by 3 matrix, where the H-function is a 3 by 3 by 3 array, since each leg has three DOF.

From Table 3-3, the first-order KIC for each leg can be expressed as shown in Table 5-1. Similarly, using Table 3-3 and the fact that

$${}_r\mathbf{P} = L_3 \begin{bmatrix} C_{\theta_1} S_{\theta_2} \\ S_{\theta_1} S_{\theta_2} \\ -C_{\theta_2} \end{bmatrix} \quad (5-16)$$

where equation 5-16 is expressed in terms of the leg frame and

$${}_r\mathbf{R}_1 = {}_r\mathbf{R}_2 = \mathbf{Q} \quad (5-17)$$

the second-order KIC are shown in Table 5-2.

This section also includes the first- and second-order KIC for the center of mass of each link to be used in the dynamic model. They are tabulated in Table 5-3 and Table 5-4, respectively. Notice that the second-order KIC for point P and the center of masses in Table 5-2 and Table 5-4 are all symmetric matrices.

#### 5-1.5 First- and Second-order Kinematics of the Platform

Recalling Section 5-1.2, the positions of the six connecting points at the ball joints are fixed with respect to the platform frame. Define the location of the ball joints in the platform frame as

$${}^{(pl)}_r\mathbf{W} = \begin{bmatrix} {}^{(pl)}_r\mathbf{W}_x \\ {}^{(pl)}_r\mathbf{W}_y \\ {}^{(pl)}_r\mathbf{W}_z \end{bmatrix} \quad (5-18)$$

where the pre-superscript (pl) denotes the platform reference frame. To rotate the vector in equation 5-18 to the base frame, simply premultiply equation 5-18 with the rotation matrix T, which is defined as

$$\mathbf{T} = \begin{bmatrix} \mathbf{a} & \mathbf{s} \times \mathbf{a} & \mathbf{s} \end{bmatrix} \quad (5-5)$$

$$\begin{bmatrix} {}_rG_{\varphi}^{12} \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} {}_rG_{\varphi}^{23} \end{bmatrix} = \begin{bmatrix} 0 & S_{\theta_1} & 0 \\ 0 & -C_{\theta_1} & 0 \\ 1 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} {}_rG_{\varphi}^{34} \end{bmatrix} = \begin{bmatrix} 0 & S_{\theta_1} & C_{\theta_1}S_{\theta_2} \\ 0 & -C_{\theta_1} & S_{\theta_1}S_{\theta_2} \\ 1 & 0 & -C_{\theta_2} \end{bmatrix}$$

$$\begin{bmatrix} {}_rG_{\varphi}^p \end{bmatrix} = \begin{bmatrix} -L_3S_{\theta_1}S_{\theta_2} & L_3C_{\theta_1}C_{\theta_2} & C_{\theta_1}S_{\theta_2} \\ L_3C_{\theta_1}S_{\theta_2} & L_3S_{\theta_1}C_{\theta_2} & S_{\theta_1}S_{\theta_2} \\ 0 & L_3S_{\theta_2} & -C_{\theta_2} \end{bmatrix}$$

Table 5-1 First-order KIC for Each Link

$$[{}^r H_{\phi}^{12}]_{i;;} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} ; i = 1, 2, 3$$

$$[{}^r H_{\phi}^{23}]_{1;;} = \begin{bmatrix} 0 & C_{\theta_1} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$[{}^r H_{\phi}^{23}]_{2;;} = \begin{bmatrix} 0 & S_{\theta_1} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$[{}^r H_{\phi}^{23}]_{3;;} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

$$[{}^r H_{\phi}^{34}]_{1;;} = \begin{bmatrix} 0 & C_{\theta_1} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$[{}^r H_{\phi}^{34}]_{2;;} = \begin{bmatrix} 0 & S_{\theta_1} & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$[{}^r H_{\phi}^{34}]_{3;;} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

Table 5-2 Second-order KIC for Each of the Links

$$[{}^r H_\phi^p]_{1;;} = \begin{bmatrix} -L_3 C_{\theta_1} S_{\theta_2} & -L_3 S_{\theta_1} C_{\theta_2} & -S_{\theta_1} S_{\theta_2} \\ -L_3 S_{\theta_1} C_{\theta_2} & -L_3 C_{\theta_1} S_{\theta_2} & C_{\theta_1} C_{\theta_2} \\ -S_{\theta_1} S_{\theta_2} & C_{\theta_1} C_{\theta_2} & 0 \end{bmatrix}$$

$$[{}^r H_\phi^p]_{2;;} = \begin{bmatrix} -L_3 S_{\theta_1} S_{\theta_2} & L_3 C_{\theta_1} C_{\theta_2} & C_{\theta_1} S_{\theta_2} \\ L_3 C_{\theta_1} C_{\theta_2} & -L_3 S_{\theta_1} S_{\theta_2} & S_{\theta_1} C_{\theta_2} \\ C_{\theta_1} S_{\theta_2} & S_{\theta_1} C_{\theta_2} & 0 \end{bmatrix}$$

$$[{}^r H_\phi^p]_{3;;} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & L_3 C_{\theta_2} & S_{\theta_2} \\ 0 & S_{\theta_2} & 0 \end{bmatrix}$$

Table 5-2 Second-order KIC for Each of the Links  
(cont.)

$$\begin{bmatrix} {}^1_r G^c_\varphi \end{bmatrix} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix}$$

$$\begin{bmatrix} {}^2_r G^c_\varphi \end{bmatrix} = \begin{bmatrix} -\frac{L_c}{2} S_{\theta_1} S_{\theta_2} & \frac{L_c}{2} C_{\theta_1} C_{\theta_2} & 0 \\ \frac{L_c}{2} C_{\theta_1} S_{\theta_2} & \frac{L_c}{2} S_{\theta_1} C_{\theta_2} & 0 \\ 0 & \frac{L_c}{2} S_{\theta_2} & 0 \end{bmatrix}$$

$$\begin{bmatrix} {}^3_r G^c_\varphi \end{bmatrix} = \begin{bmatrix} -L S_{\theta_1} S_{\theta_2} & L C_{\theta_1} C_{\theta_2} & C_{\theta_1} S_{\theta_2} \\ L C_{\theta_1} S_{\theta_2} & L S_{\theta_1} C_{\theta_2} & S_{\theta_1} S_{\theta_2} \\ 0 & L S_{\theta_2} & -C_{\theta_2} \end{bmatrix}$$

where  $L_c$  is the length from the origin of the link frame to the center of mass of the piston cylinder. And

$$L = L_3 - \frac{L_r}{2}$$

with  $L_3$  defined as the total length of the prismatic link, and  $L_r$  is the piston rod length.

**Table 5-3 First-order KIC for Each of the Center of Masses**

$$\left[ {}^1H_{\varphi\varphi}^c \right]_{i;;} = \begin{bmatrix} 0 & 0 & 0 \end{bmatrix} \quad ; \quad i = 1, 2, 3$$

$$\left[ {}^2H_{\varphi\varphi}^c \right]_{1;;} = \begin{bmatrix} -\frac{L_c}{2} C_{\theta_1} S_{\theta_2} & -\frac{L_c}{2} S_{\theta_1} C_{\theta_2} & 0 \\ -\frac{L_c}{2} S_{\theta_1} C_{\theta_2} & -\frac{L_c}{2} C_{\theta_1} S_{\theta_2} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\left[ {}^2H_{\varphi\varphi}^c \right]_{2;;} = \begin{bmatrix} -\frac{L_c}{2} S_{\theta_1} S_{\theta_2} & \frac{L_c}{2} C_{\theta_1} C_{\theta_2} & 0 \\ \frac{L_c}{2} C_{\theta_1} C_{\theta_2} & -\frac{L_c}{2} S_{\theta_1} S_{\theta_2} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

$$\left[ {}^2H_{\varphi\varphi}^c \right]_{3;;} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & \frac{L_c}{2} C_{\theta_2} & 0 \\ 0 & 0 & 0 \end{bmatrix}$$

Table 5-4 Second-order KIC for Each of the Center of Masses

$$\left[ {}^3H_{\varphi\varphi}^c \right]_{1;;} = \begin{bmatrix} -L C_{\theta_1} S_{\theta_2} & -L S_{\theta_1} C_{\theta_2} & -S_{\theta_1} S_{\theta_2} \\ -L S_{\theta_1} C_{\theta_2} & -L C_{\theta_1} S_{\theta_2} & C_{\theta_1} C_{\theta_2} \\ -S_{\theta_1} S_{\theta_2} & C_{\theta_1} C_{\theta_2} & 0 \end{bmatrix}$$

$$\left[ {}^3H_{\varphi\varphi}^c \right]_{2;;} = \begin{bmatrix} -L S_{\theta_1} S_{\theta_2} & L C_{\theta_1} C_{\theta_2} & C_{\theta_1} S_{\theta_2} \\ L C_{\theta_1} C_{\theta_2} & -L S_{\theta_1} S_{\theta_2} & S_{\theta_1} C_{\theta_2} \\ C_{\theta_1} S_{\theta_2} & S_{\theta_1} C_{\theta_2} & 0 \end{bmatrix}$$

$$\left[ {}^3H_{\varphi\varphi}^c \right]_{3;;} = \begin{bmatrix} 0 & 0 & 0 \\ 0 & C_{\theta_2} & S_{\theta_2} \\ 0 & S_{\theta_2} & 0 \end{bmatrix}$$

Table 5-4 Second-order KIC for Each of the Center of Masses (cont.)

Letting

$$\underline{e} = \underline{s} \times \underline{a} \quad (5-19)$$

and substituting into equation 5-5 yields

$$T = \begin{bmatrix} \underline{a} & \underline{e} & \underline{s} \end{bmatrix} = \begin{bmatrix} \dot{X}_a & \dot{X}_e & \dot{X}_s \\ \dot{Y}_a & \dot{Y}_e & \dot{Y}_s \\ \dot{Z}_a & \dot{Z}_e & \dot{Z}_s \end{bmatrix} \quad (5-20)$$

where the first column is the unit vector of the platform x-axis, the second column is the y-axis and the last column is the z-axis, all expressed in the base coordinates frame, implied by the superscript (\*). Thus

$${}_r\underline{W} = T^{(pl)}{}_r\underline{W} \quad (5-21)$$

or

$$\begin{bmatrix} {}_rW_x \\ {}_rW_y \\ {}_rW_z \end{bmatrix} = \begin{bmatrix} X_a^{(pl)}{}_rW_x + X_e^{(pl)}{}_rW_y + X_s^{(pl)}{}_rW_z \\ Y_a^{(pl)}{}_rW_x + Y_e^{(pl)}{}_rW_y + Y_s^{(pl)}{}_rW_z \\ Z_a^{(pl)}{}_rW_x + Z_e^{(pl)}{}_rW_y + Z_s^{(pl)}{}_rW_z \end{bmatrix} \quad (5-22)$$

To determine the G- and H-functions for the ball joints in terms of platform coordinates (which will be necessary to determine the G- and H-functions relating the joint coordinates  ${}^r\mathcal{Q}$  to the platform coordinates  $\underline{U}$ ), the velocity and acceleration of the point P at the ball joint of leg r are written as

$${}^r\dot{P} = \dot{\underline{U}} + \underline{\omega} \times {}_r\underline{W} \quad (5-23)$$

and

$${}^r\ddot{P} = \ddot{\underline{U}} + \underline{\alpha} \times {}_r\underline{W} + \underline{\omega} \times (\underline{\omega} \times {}_r\underline{W}) \quad (5-24)$$

where  $\underline{U}$  is the position vector of the origin of the platform frame expressed in the base frame, and  $\underline{\omega}$  and  $\underline{\alpha}$  are the angular velocity and acceleration of the platform, respectively. From equations 5-23 and 5-24, the first-order KIC are

$$\begin{bmatrix} {}^rG_u^p \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & {}^rW_z & -{}^rW_y \\ 0 & 1 & 0 & -{}^rW_z & 0 & {}^rW_x \\ 0 & 0 & 1 & {}^rW_y & -{}^rW_x & 0 \end{bmatrix} \quad (5-25)$$

The components in the three planes of the H-function are zero except the following

$$\begin{bmatrix} {}^rH_{uu}^p \end{bmatrix}_{1;m;n} = \begin{bmatrix} 0 & {}^rW_y & {}^rW_z \\ 0 & -{}^rW_x & 0 \\ 0 & 0 & -{}^rW_x \end{bmatrix} \quad (5-26)$$

$$\begin{bmatrix} {}^rH_{uu}^p \end{bmatrix}_{2;m;n} = \begin{bmatrix} -{}^rW_y & {}^rW_x & 0 \\ 0 & 0 & {}^rW_z \\ 0 & 0 & -{}^rW_y \end{bmatrix} \quad (5-27)$$

$$\begin{bmatrix} {}^rH_{uu}^p \end{bmatrix}_{3;m;n} = \begin{bmatrix} -{}^rW_z & 0 & {}^rW_x \\ 0 & -{}^rW_z & {}^rW_y \\ 0 & 0 & 0 \end{bmatrix} \quad (5-28)$$

where  $m, n = 4, 5, 6$ . As a result, the velocity and acceleration can be written as

$${}^r\dot{P} = \begin{bmatrix} {}^rG_u^p \end{bmatrix} \dot{\underline{U}} \quad (5-29)$$

$${}^r\ddot{P} = \begin{bmatrix} {}^rG_u^p \end{bmatrix} \ddot{\underline{U}} + {}^r\dot{P}^T \begin{bmatrix} {}^rH_{uu}^p \end{bmatrix} {}^r\dot{P} \quad (5-30)$$

in terms of the G- and H-functions.

This concludes the kinematic modeling of the DDTS(or the generalized Stewart Platform) referenced to the initial sets of generalized coordinates(ie.,  ${}^r\dot{Q}$  and  $\underline{U}$ ). One very important point to note is that the derivation so far expresses the

G- and H-functions for P in terms of  $\Phi$  in the leg frame, whereas, the G- and H-functions for P are in terms of  $\underline{U}$  in the base frame.

## 5-2 Initial Dynamic Model of the Legs

Similar to the kinematic modeling discussed in the previous section, the initial dynamic model also considers one of the legs and treats it as if it is an isolated three DOF serial manipulator. Assuming that there is only inertial loading, the procedures to arrive at the initial dynamic model follow the derivation of Section 3-3.2. Referring to Fig. 5-2, the model here assumes that the hooke joint is a solid cylinder, the piston cylinder and the piston rod are considered slender rods. Due to the symmetric nature of each of the links in the leg, all the local inertia tensors are diagonal matrices.

Substituting the appropriate terms into equation 3-55, the leg frame referenced inertia tensor for each of the links in Fig. 5-2 are given in Table 5-5. The symbol I stands for the mass moment of inertia of each links, where

$$\begin{array}{ll} I_{x1} \text{ is along } \underline{a}_{12} & ; \quad I_{z1} \text{ is along } \underline{s}_1 \\ I_{x2} \text{ is along } \underline{a}_{23} & ; \quad I_{z2} \text{ is along } \underline{s}_2 \\ I_{x3} \text{ is along } \underline{a}_{34} & ; \quad I_{z3} \text{ is along } \underline{s}_3 \end{array}$$

Substituting the inertia matrices in Table 5-5 and the G- and H-functions determined in Section 5-1 into the effective inertia matrix equation 3-66 and the inertia power array equation 3-67 results in the model coefficients tabulated in Table 5-6 and Table 5-7, respectively. Finally, having determined  $[\text{}^r\dot{I}_{\Phi\Phi}]$  and  $[\text{}^r\dot{P}_{\Phi\Phi\Phi}]$ , substitution into equation 3-66, gives the initial inertial load expression for the dynamic model of each leg as

$$\text{}^r\mathbf{I}_{\Phi}^I = [\text{}^r\dot{I}_{\Phi\Phi}] \ddot{\Phi} + \dot{\Phi}^T [\text{}^r\dot{P}_{\Phi\Phi\Phi}] \dot{\Phi} \quad (5-31)$$

$$[{}_r\Pi_{12}] = \begin{bmatrix} I_{x1}C_{\theta_1}^2 + I_{y1}S_{\theta_1}^2 & (I_{x1} - I_{y1})S_{\theta_1}C_{\theta_1} & 0 \\ (I_{x1} - I_{y1})S_{\theta_1}C_{\theta_1} & I_{x1}S_{\theta_1}^2 + I_{y1}C_{\theta_1}^2 & 0 \\ 0 & 0 & I_{z1} \end{bmatrix}$$

$$[{}_r\Pi_{23}] = \begin{bmatrix} I_{x2}C_{\theta_1}^2C_{\theta_2}^2 + I_{z1}S_{\theta_1}^2 & I_{x2}S_{\theta_1}C_{\theta_1}C_{\theta_2}^2 - I_{z2}S_{\theta_1}C_{\theta_1} & \\ I_{x2}S_{\theta_1}C_{\theta_1}C_{\theta_2}^2 - I_{z2}S_{\theta_1}C_{\theta_1} & I_{x2}S_{\theta_1}^2C_{\theta_2}^2 + I_{z2}C_{\theta_1}^2 & \\ I_{x2}C_{\theta_1}C_{\theta_2}S_{\theta_2} & I_{x2}S_{\theta_1}S_{\theta_2}C_{\theta_2} & \\ & & I_{x2}C_{\theta_1}C_{\theta_2}S_{\theta_2} \\ & & I_{x2}S_{\theta_1}S_{\theta_2}C_{\theta_2} \\ & & I_{x2}S_{\theta_2}^2 \end{bmatrix}$$

$$[{}_r\Pi_{34}] = \begin{bmatrix} I_{x3}C_{\theta_1}^2C_{\theta_2}^2 + I_{y3}S_{\theta_1}^2 & I_{x3}S_{\theta_1}C_{\theta_1}C_{\theta_2}^2 - I_{y3}S_{\theta_1}C_{\theta_1} & \\ I_{x3}S_{\theta_1}C_{\theta_1}C_{\theta_2}^2 - I_{y2}S_{\theta_1}C_{\theta_1} & I_{x3}S_{\theta_1}^2C_{\theta_2}^2 + I_{y3}C_{\theta_1}^2 & \\ I_{x3}C_{\theta_1}C_{\theta_2}S_{\theta_2} & I_{x3}S_{\theta_1}S_{\theta_2}C_{\theta_2} & \\ & & I_{x3}C_{\theta_1}C_{\theta_2}S_{\theta_2} \\ & & I_{x3}S_{\theta_1}S_{\theta_2}C_{\theta_2} \\ & & I_{x3}S_{\theta_2}^2 \end{bmatrix}$$

Table 5-5 Inertia Tensors

$$\begin{bmatrix} {}_r\dot{I}_{\varphi\varphi} \end{bmatrix} = \begin{bmatrix} A & 0 & 0 \\ 0 & B & 0 \\ 0 & 0 & M_{34} \end{bmatrix}$$

where

$$A = I_{z1} + M_{23} \left( \frac{L_c}{2} S_{\theta_2} \right)^2 + M_{34} L^2 S_{\theta_2}^2 + I_{x2} S_{\theta_2}^2 + I_{x3} S_{\theta_2}^2$$

$$B = I_{z2} + I_{y3} + M_{23} \left( \frac{L_c}{2} \right)^2 + M_{34} L^2$$

and

$$L = L_3 - \frac{L_r}{2}$$

**Table 5-6 Effective Inertia Matrix**

$$\begin{aligned}
\left[ {}_r P_{\phi\phi\phi}^{\cdot} \right]_{1;;} &= \begin{bmatrix} 0 & D & M_{34} L S_{\theta_2}^2 \\ C & 0 & -I_{y3} S_{\theta_2} \\ M_{34} L S_{\theta_2}^2 & 0 & 0 \end{bmatrix} \\
\left[ {}_r P_{\phi\phi\phi}^{\cdot} \right]_{2;;} &= \begin{bmatrix} E & 0 & I_{x3} S_{\theta_2} \\ 0 & 0 & M_{34} L \\ 0 & M_{34} L & 0 \end{bmatrix} \\
\left[ {}_r P_{\phi\phi\phi}^{\cdot} \right]_{3;;} &= \begin{bmatrix} -M_{34} L S_{\theta_2}^2 & I_{x3} S_{\theta_2} & 0 \\ I_{y3} S_{\theta_2} & -M_{34} L & 0 \\ 0 & 0 & 0 \end{bmatrix}
\end{aligned}$$

where

$$C = \left( M_{23} \left( \frac{L_c}{2} \right)^2 + M_{34} L^2 \right) S_{\theta_2} C_{\theta_2}$$

$$D = \left( M_{23} \left( \frac{L_c}{2} \right)^2 + M_{34} L^2 + 2 I_{x2} + 2 I_{x3} \right) S_{\theta_2} C_{\theta_2}$$

$$E = \left( -M_{23} \left( \frac{L_c}{2} \right)^2 - M_{34} L^2 - I_{x2} - I_{x3} \right) S_{\theta_2} C_{\theta_2}$$

and

$$L = L_3 - \frac{L_r}{2}$$

Table 5-7 Inertia Power Array

### 5-3 Transfer of Generalized Coordinates to Obtain the Dynamic Model Referenced to the Platform

Section 5-2 developed the initial inertial load model of each leg directly in terms of the relative joint parameters  ${}^r\Phi$ . Since it is assumed that there is no external load applied to the system, the controlling equation of motion in equation 5-31 can be written as

$$\begin{aligned} {}^r\mathbf{I}_\Phi &= {}^r\mathbf{I}_\Phi^I \\ &= \left[ {}^r\mathbf{I}_{\Phi\Phi}^I \right] \ddot{\Phi} + \dot{\Phi}^T \left[ {}^r\mathbf{P}_{\Phi\Phi\Phi}^I \right] \dot{\Phi} \end{aligned} \quad (5-32)$$

Before arriving at the desired model, which will be referenced to the actuators, it is necessary to transfer the dynamic model in equation 5-32 to an intermediate set of generalized coordinates, which in this case is the platform coordinate set  $\mathbf{U}$ . Using the relations

$$\dot{\Phi} = \left[ {}^r\mathbf{G}_\Phi^p \right]^{-1} {}^r\dot{\mathbf{P}} \quad (5-33)$$

$$\ddot{\Phi} = \left[ {}^r\mathbf{G}_\Phi^p \right]^{-1} {}^r\ddot{\mathbf{P}} + \dot{\mathbf{P}}^T \left[ {}^r\mathbf{H}_{pp}^\Phi \right] {}^r\dot{\mathbf{P}} \quad (5-34)$$

$${}^r\dot{\mathbf{P}} = \left[ {}^r\mathbf{G}_u^p \right] \dot{\mathbf{U}} \quad (5-35)$$

and

$${}^r\ddot{\mathbf{P}} = \left[ {}^r\mathbf{G}_u^p \right] \ddot{\mathbf{U}} + \dot{\mathbf{U}}^T \left[ {}^r\mathbf{H}_{uu}^p \right] \dot{\mathbf{U}} \quad (5-36)$$

equation 5-32 can be expressed as

$${}^r\mathbf{I}_\Phi = \left[ {}^r\mathbf{I}_{\Phi\Phi}^I \right] \left[ {}^r\mathbf{G}_\Phi^p \right]^{-1} \left[ {}^r\mathbf{G}_u^p \right] \ddot{\mathbf{U}} + \dot{\mathbf{U}}^T \left[ {}^r\mathbf{P}_{\Phi uu} \right] \dot{\mathbf{U}} \quad (5-37)$$

where

$$\begin{aligned}
[{}^r P_{\phi uu}] &= [{}^r G_u^p]^T \left( [{}^r G_\phi^p]^T [{}^r P_{\phi\phi\phi}^\bullet] [{}^r G_\phi^p]^{-1} \right. \\
&\quad \left. + [{}^r I_{\phi\phi}^\bullet] \cdot [{}^r H_{pp}^\phi] \right) [{}^r G_u^p] + \left( [{}^r I_{\phi\phi}^\bullet] [{}^r G_\phi^p]^{-1} \right) \cdot [{}^r H_{uu}^p]
\end{aligned}
\tag{5-38}$$

Also, substituting equations 5-35 and 5-36 into equation 5-34 for  ${}^r \ddot{P}$  and  ${}^r \ddot{P}$  gives

$$\begin{aligned}
\ddot{\Phi} &= [{}^r G_\phi^p]^{-1} \left( [{}^r G_u^p] \dot{U} + \dot{U}^T [{}^r H_{uu}^p] \dot{U} \right) \\
&\quad + \dot{U}^T \left( [{}^r G_u^p]^T [{}^r H_{pp}^\phi] [{}^r G_u^p] \right) \dot{U}
\end{aligned}
\tag{5-39}$$

the G- and H-functions for the joints referenced to the set of generalized coordinates associated with the platform can then be expressed as

$$[{}^r G_u^\phi] = [{}^r G_\phi^p]^{-1} [{}^r G_u^p] \tag{5-40}$$

and

$$[{}^r H_{uu}^\phi] = [{}^r G_\phi^p]^{-1} \cdot [{}^r H_{uu}^p] + [{}^r G_u^p]^T [{}^r H_{pp}^\phi] [{}^r G_u^p] \tag{5-41}$$

Note, the kinematic influence coefficients relating the initial coordinate sets  ${}^r \Phi$  to the platform coordinates  $U$ , which are required to determine the platform based model, cannot be found by direct inversion of those relating  $U$  to  ${}^r \Phi$  as in Hudgens [21], since they do not exist. Here,  ${}^r \Phi$  is related to  $P$  and  $P$  is related to  $U$  yielding the required relations of  ${}^r \Phi$  to  $U$  as given by the above equations. Next, utilizing the relation

$${}^r I_u = [{}^r G_\phi^u]^T {}^r I_\phi \tag{5-42}$$

the 6 by 6 effective inertia matrix and the 6 by 6 by 6 inertia power array referenced to the platform set of generalized coordinates can then be expressed as

$$\begin{bmatrix} \dot{I}_{uu} \end{bmatrix} = \begin{bmatrix} {}^r G_u^p \end{bmatrix}^T \begin{bmatrix} {}^r G_\phi^p \end{bmatrix}^{-T} \begin{bmatrix} {}^r I_{\phi\phi} \end{bmatrix} \begin{bmatrix} {}^r G_\phi^p \end{bmatrix}^{-1} \begin{bmatrix} {}^r G_u^p \end{bmatrix} \quad (5-43)$$

$$\begin{bmatrix} \dot{P}_{uuu} \end{bmatrix} = \begin{bmatrix} {}^r G_u^\phi \end{bmatrix}^T \cdot \begin{bmatrix} {}^r P_{\phi uu} \end{bmatrix} \quad (5-44)$$

where

$$\begin{bmatrix} {}^r G_u^\phi \end{bmatrix} = \begin{bmatrix} {}^r G_p^p \end{bmatrix} \begin{bmatrix} {}^r G_u^p \end{bmatrix} \quad (5-45)$$

Now that the dynamic model for each leg is referenced to the common platform generalized coordinates, they can be combined by substituting into the following equations similar to equations 4-42 and 4-43.

$$\begin{bmatrix} \dot{I}_{uu} \end{bmatrix} = \begin{bmatrix} I_{uu} \end{bmatrix} + \sum_{r=1}^6 \begin{bmatrix} {}^r \dot{I}_{uu} \end{bmatrix} \quad (5-46)$$

$$\begin{bmatrix} \dot{P}_{uuu} \end{bmatrix} = \begin{bmatrix} P_{uu} \end{bmatrix} + \sum_{r=1}^6 \begin{bmatrix} {}^r \dot{P}_{uuu} \end{bmatrix} \quad (5-47)$$

Finally, to complete the model of the DDTS, the platform effective inertia matrix and its inertia power array need to be determined and substituted into the above equations. They can be expressed as

$$\begin{bmatrix} I_{uu} \end{bmatrix} = \begin{bmatrix} M_{pl} & 0 & 0 & Q^T \\ 0 & M_{pl} & 0 & Q^T \\ 0 & 0 & M_{pl} & Q^T \\ Q & Q & Q & \begin{bmatrix} \Pi^{pl} \end{bmatrix} \end{bmatrix} \quad (5-48)$$

$$\begin{bmatrix} P_{uuu} \end{bmatrix}_{k;;} = \begin{bmatrix} Q & Q & Q & Q & Q & Q \\ Q & Q & Q & \begin{bmatrix} P_{uuu} \end{bmatrix}_{k;m;n} \end{bmatrix} ; k, m, n = 4, 5, 6 \quad (5-49)$$

where  $M_{pl}$  is the platform mass and the inertia tensor is

$$\begin{aligned}
 [\Pi^{pl}] &= \left[ \begin{array}{c|c|c} \dot{I}^x & \dot{I}^y & \dot{I}^z \end{array} \right] \\
 &= T \left[ \begin{array}{c} (pl) \Pi^{pl} \end{array} \right] T^T
 \end{aligned} \tag{5-50}$$

with the local inertia tensor being

$$\left[ \begin{array}{c} (pl) \Pi^{pl} \end{array} \right] = \left[ \begin{array}{ccc} \frac{M_{pl} W^2}{4} & 0 & 0 \\ 0 & \frac{M_{pl} W^2}{4} & 0 \\ 0 & 0 & \frac{M_{pl} W^2}{2} \end{array} \right] \tag{5-51}$$

where  $W$  is the radius of the platform, which is the magnitude of the vector in equation 5-21. Note that the first three planes of  $[P_{uuu}]$  are 6 by 6 null matrices, and

$$[P_{uuu}]_{4;m;n} = \left[ \begin{array}{c|c|c} 0 & \dot{I}^z & -\dot{I}^y \end{array} \right] \tag{5-52}$$

$$[P_{uuu}]_{5;m;n} = \left[ \begin{array}{c|c|c} -\dot{I}^z & 0 & \dot{I}^x \end{array} \right] \tag{5-53}$$

$$[P_{uuu}]_{6;m;n} = \left[ \begin{array}{c|c|c} \dot{I}^y & -\dot{I}^x & 0 \end{array} \right] \tag{5-54}$$

where  $m, n = 4, 5, 6$ .

As a result of the above transfer of generalized coordinates, the complete dynamic model of the DDTS is now referenced to the common set of platform coordinates. The next section will discuss the final transfer of system dependence to the desired set of generalized coordinates.

5-4 Transfer of the Dynamic Model to the Desired Set of Generalized Coordinates--The Input Actuators

Having the complete model of the system referenced to the platform coordinates, the final step is to transfer the model to each of the actuators. Since the actuator in each leg is at the third joint, the desired set of first-order KIC can be obtained by merely extracting the third row from the result of equation 5-40 for each of the legs, ie.,

$$\begin{bmatrix} G_u^d \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} {}^1G_u^\varphi \end{bmatrix}_{3;} \\ \begin{bmatrix} {}^2G_u^\varphi \end{bmatrix}_{3;} \\ \begin{bmatrix} {}^3G_u^\varphi \end{bmatrix}_{3;} \\ \begin{bmatrix} {}^4G_u^\varphi \end{bmatrix}_{3;} \\ \begin{bmatrix} {}^5G_u^\varphi \end{bmatrix}_{3;} \\ \begin{bmatrix} {}^6G_u^\varphi \end{bmatrix}_{3;} \end{bmatrix} \quad (5-55)$$

Similarly, the second-order KIC is obtained by extracting the third plane from the result of equation 5-41 for each of the legs, ie.,

$$\begin{bmatrix} H_{uu}^d \end{bmatrix} = \begin{bmatrix} \begin{bmatrix} {}^1H_{uu}^\varphi \end{bmatrix}_{3;;} \\ \begin{bmatrix} {}^2H_{uu}^\varphi \end{bmatrix}_{3;;} \\ \begin{bmatrix} {}^3H_{uu}^\varphi \end{bmatrix}_{3;;} \\ \begin{bmatrix} {}^4H_{uu}^\varphi \end{bmatrix}_{3;;} \\ \begin{bmatrix} {}^5H_{uu}^\varphi \end{bmatrix}_{3;;} \\ \begin{bmatrix} {}^6H_{uu}^\varphi \end{bmatrix}_{3;;} \end{bmatrix} \quad (5-56)$$

This extraction procedure has been described in Section 4-3. Once equations 5-55 and 5-56 are determined, the effective inertia matrix and inertia power array can easily be computed by recalling equations 4-55 and 4-56 as

$$[\dot{I}_{dd}] = [G_u^d]^{-T} [\dot{I}_{uu}] [G_u^d]^{-1} \quad (5-57)$$

and

$$[\dot{P}_{ddd}] = [G_u^d]^{-T} \left\{ \left( [G_u^d]^{-T} \cdot [\dot{P}_{uuu}] \right) - \left( [\dot{I}_{dd}] \cdot [H_{uu}^d] \right) \right\} [G_u^d]^{-1} \quad (5-58)$$

Finally, substituting equations 5-57 and 5-58 into the dynamic equation

$$\mathbf{I}_d = [\dot{I}_{dd}] \ddot{\mathbf{d}} + \dot{\mathbf{d}}^T [\dot{P}_{ddd}] \dot{\mathbf{d}} \quad (5-59)$$

gives the necessary generalized force at each of the actuators  $\mathbf{d}$ , where

$$\mathbf{d} = \begin{bmatrix} {}_1S_3 \\ {}_2S_3 \\ {}_3S_3 \\ {}_4S_3 \\ {}_5S_3 \\ {}_6S_3 \end{bmatrix} \quad (5-60)$$

This concludes the dynamic modeling of the generalized Stewart Platform(or the DDTS) needed for the simulations addressed in the next chapter. Again, note that the final model essentially results from the multiple application of the isomorphic transformation equations to simple open-chain models. This avoids the much more difficult task of determining the desired dynamic model directly in terms of the desired generalized input coordinates.

## CHAPTER 6

### APPLICATION OF THE MODEL

#### 6-1 Verification of the Model

The complete model of the generalized Stewart Platform(or the DDTS) was developed in Chapter 5. The final model is used to calculate the generalized forces for each input, given the position, velocity and acceleration of the inputs, which are determined once the platform motion is specified. This section will discuss three different approaches employed to verify the modeling technique, particularly the distribution of the system mass parameters introduced in Chapter 5. Additional verification of the general modeling technique can be found in Freeman and Tesar [14].

##### 6-1.1 Special Case Model

The first verification procedure involves specifying a system in which the desired model coefficients are available by inspection. Applying the transfer procedure and comparing those results with the known coefficient values completes the verification process.

Instead of having the third(or the prismatic) joint from each leg be the input locations as in Section 5-4, the first joint, which is the fixed revolute of the hooke joint, is specified as the desired input location for each leg. Thus,

$$\underline{d} = \left[ \begin{matrix} {}_1\phi_1 & {}_2\phi_1 & {}_3\phi_1 & {}_4\phi_1 & {}_5\phi_1 & {}_6\phi_1 \end{matrix} \right]^T \quad (6-1)$$

is the desired input set. Furthermore, only the first joint from each leg is given mass so as to have direct knowledge of the desired model. Hence, substituting  ${}^rM_{12} = 10.0\text{kg}$  and the radius of the hooke joint as  $r = 0.06\text{m}$  into Table 5-6, gives the effective inertia matrix as

$$[{}^r\dot{I}_{\phi\phi}] = \begin{bmatrix} 0.018 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix} \quad (6-2)$$

where

$$A = I_{z1} = \frac{{}^rM_{12}r^2}{2} = 0.018 \text{ kg.m}^2 \quad (6-3)$$

Note that the pre-subscript  $r$  in  ${}^rM_{12}$  refers to the leg number and the  $r$  in  $r^2$  refers to the radius of the hooke joint.

Following these assumptions, by inspection, the effective inertia matrix referenced to the desired set of coordinates (equation 6-1) is

$$[I_{dd}^*] = \begin{bmatrix} 0.018 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.018 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.018 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.018 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.018 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.018 \end{bmatrix} \quad (6-4)$$

Also, the inertia power array  $[P_{ddd}^*]$  is a 6 by 6 by 6 null array (ie., each actuator is only responsible for its own mass).

Having knowledge of the expected results for  $[I_{dd}^*]$  and  $[P_{ddd}^*]$ , the modeling procedure outlined in Chapter 5 is then applied. First, the directly available joint referenced model coefficients for each leg,

$$[{}^rG_{\phi}^p], [{}^rH_{\phi\phi}^p], [{}^r\dot{I}_{\phi\phi}], [{}^rP_{\phi\phi\phi}^*] ; r = 1, 2, \dots, 6 \quad (6-5)$$

are determined. Second, applying the transfer of generalized coordinates to the results in equation 6-3 to obtain the model referenced to the common set of platform coordinates (equations 5-40 to 5-44) yields

$$[{}^rG_u^q], [{}^rH_{uu}^q], [{}^r\dot{I}_{uu}], [{}^rP_{uuu}^*] ; r = 1, 2, \dots, 6 \quad (6-6)$$

Since the platform is assumed to be massless, equations 5-46 and 5-47 become

$$[\dot{I}_{uu}] = \sum_{r=1}^6 [{}^r\dot{I}_{uu}] \quad (6-7)$$

and

$$[\dot{P}_{uuu}] = \sum_{r=1}^6 [{}^r\dot{P}_{uuu}] \quad (6-8)$$

Equation 6-7 gives

$$[\dot{I}_{uu}] = \begin{bmatrix} 0.0020 & 0 & 0 & -0.0023 & 0 & 0 \\ 0 & 0.0029 & 0 & 0 & 0.0011 & -0.0012 \\ 0 & 0 & 0.0029 & 0 & 0.0012 & 0.0011 \\ -0.0023 & 0 & 0 & 0.0030 & 0 & 0 \\ 0 & 0.0011 & 0.0012 & 0 & 0.0039 & 0 \\ 0 & -0.0012 & 0.0011 & 0 & 0 & 0.0039 \end{bmatrix} \quad (6-9)$$

which is a symmetric matrix. Then, performing the extraction procedure discussed in Section 4-3 to the first- and second-order KIC in equation 6-4 to obtain

$$[G_u^d] = \begin{bmatrix} [{}^1G_u^\Phi]_{1;} \\ [{}^2G_u^\Phi]_{1;} \\ [{}^3G_u^\Phi]_{1;} \\ [{}^4G_u^\Phi]_{1;} \\ [{}^5G_u^\Phi]_{1;} \\ [{}^6G_u^\Phi]_{1;} \end{bmatrix} \quad (6-10)$$

and

$$[H_{uu}^d] = \begin{bmatrix} [{}^1H_{uu}^\varphi]_{1;;} \\ [{}^2H_{uu}^\varphi]_{1;;} \\ [{}^3H_{uu}^\varphi]_{1;;} \\ [{}^4H_{uu}^\varphi]_{1;;} \\ [{}^5H_{uu}^\varphi]_{1;;} \\ [{}^6H_{uu}^\varphi]_{1;;} \end{bmatrix} \quad (6-11)$$

as the G- and H-functions relating the desired input coordinates(d) to the common platform coordinates(U). Finally, applying the transfer equations again one arrives at the desired model coefficients(equations 5-57 and 5-58),

$$[\dot{I}_{dd}] = [G_u^d]^{-T} [\dot{I}_{uu}] [G_u^d]^{-1} \quad (6-12)$$

and

$$[P_{ddd}^\bullet] = [G_u^d]^{-T} \left\{ \left( [G_u^d]^{-T} \cdot [P_{uuu}^\bullet] \right) - \left( [\dot{I}_{dd}] \cdot [H_{uu}^d] \right) \right\} [G_u^d]^{-1} \quad (6-13)$$

Substituting equation 6-9 and the result from equation 6-8 into equations 6-12 and 6-13 yield

$$[\dot{I}_{dd}] = \begin{bmatrix} 0.018 & 0 & 0 & 0 & 0 & 0 \\ 0 & 0.018 & 0 & 0 & 0 & 0 \\ 0 & 0 & 0.018 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0.018 & 0 & 0 \\ 0 & 0 & 0 & 0 & 0.018 & 0 \\ 0 & 0 & 0 & 0 & 0 & 0.018 \end{bmatrix} \quad (6-14)$$

and

$$[P_{ddd}^\bullet]_{k;;} = [0] ; \quad k = 1, 2, \dots, 6 \quad (6-15)$$

where each plane in equation 6-15 is a 6 by 6 matrix. Comparison of the results of equations 6-14 and 6-15 with the known results partially verifies the model

developed in Chapter 5 and the computer program written for the simulation in the later part of this chapter.

### 6-1.2 Verification of the Result for the First-order KIC

One quick and simple way to check part of the model is by verifying the first-order KIC. From Table 3-3, the directly obtained first-order KIC can be expressed as

$$\begin{aligned} \begin{bmatrix} {}^rG_{\phi}^p \end{bmatrix} &= \begin{bmatrix} \underline{s}_1 \times ({}^rP - R_1) & \underline{s}_2 \times ({}^rP - R_2) & \underline{s}_3 \end{bmatrix} \\ &= \begin{bmatrix} a \underline{s}_2 & b (\underline{s}_2 \times \underline{s}_3) & \underline{s}_3 \end{bmatrix} \end{aligned} \quad (6-16)$$

where  $a$  and  $b$  are some constants, and from equation 5-25

$$\begin{bmatrix} {}^rG_u^p \end{bmatrix} = \begin{bmatrix} 1 & 0 & 0 & 0 & {}_rW_z & -{}_rW_y \\ 0 & 1 & 0 & -{}_rW_z & 0 & {}_rW_x \\ 0 & 0 & 1 & {}_rW_y & -{}_rW_x & 0 \end{bmatrix} \quad (6-17)$$

are the platform referenced G-functions. Using the transfer of generalized coordinates and realizing that only the third column of  $\begin{bmatrix} {}^rG_u^p \end{bmatrix}$  is needed since each leg's actuator is located at the third joint(prismatic), equation 5-45 can be written as

$$\begin{bmatrix} {}^rG_u^p \end{bmatrix}_{3;} = \begin{bmatrix} {}_rG_p^p \end{bmatrix}_{3;} \begin{bmatrix} {}^rG_u^p \end{bmatrix} \quad (6-18)$$

Furthermore, since

$$\begin{bmatrix} {}_rG_p^p \end{bmatrix} = \begin{bmatrix} {}^rG_{\phi}^p \end{bmatrix}^{-1} \quad (6-19)$$

the third row in the above expression can be shown to be

$$\left[ {}^r G_p^{\phi} \right]_{3;} = \underline{S}_3^T \quad (6-20)$$

Rewriting equation 6-18 gives

$$\left( \left[ {}^r G_u^{\phi} \right]_{3;} \right)^T = \left[ {}^r G_u^p \right]^T \left( \left[ {}^r G_p^{\phi} \right]_{3;} \right)^T \quad (6-21)$$

Substituting equation 6-20 into equation 6-21 yields

$$\left( \left[ {}^r G_u^{\phi} \right]_{3;} \right)^T = \begin{pmatrix} \underline{S}_3 \\ {}^r \underline{W} \times \underline{S}_3 \end{pmatrix} \quad (6-22)$$

where

$${}^r \underline{W} = \begin{bmatrix} {}^r \underline{W}_x \\ {}^r \underline{W}_y \\ {}^r \underline{W}_z \end{bmatrix} \quad (6-23)$$

which is as previously determined in equation 5-21.

Since equation 6-18 relates the directly obtainable initial G-functions to the G-functions needed for the final transfer to the desired model, the above procedure effectively checks the validity of the results for the G-functions used in the desired model.

### 6-1.3 Actuator Motion Verification

Another way to partially verify the model is by checking the first- and second-order KIC used throughout the process. The method adopted here is to compare the result determined via the model developed in Chapter 5 with that obtained from direct mathematical calculation. This method also serves as a check for the verification process described in Section 6-1.2.

Recalling equation 5-59, the terms  $\dot{\underline{d}}$  and  $\ddot{\underline{d}}$  are the velocity and acceleration of the actuators (ie., the prismatic links) in this model. As seen

throughout the development in Chapter 5, the general expression for the velocity and acceleration of the linear actuators can be written as

$$\dot{\mathbf{d}} = \begin{bmatrix} \mathbf{G}_u^d \end{bmatrix} \dot{\mathbf{U}} \quad (6-24)$$

$$\ddot{\mathbf{d}} = \begin{bmatrix} \mathbf{G}_u^d \end{bmatrix} \ddot{\mathbf{U}} + \dot{\mathbf{U}}^T \begin{bmatrix} \mathbf{H}_{uu}^d \end{bmatrix} \dot{\mathbf{U}} \quad (6-25)$$

However, the velocity and acceleration can also be determined directly by taking the time derivative of equation 5-8, ie.,

$$L_3^2 = X_p^2 + Y_p^2 + Z_p^2 \quad (6-26)$$

to obtain the velocity expression as

$$\dot{L}_3 = \frac{X_p \dot{X}_p + Y_p \dot{Y}_p + Z_p \dot{Z}_p}{L_3} \quad (6-27)$$

Taking the time derivative of equation 6-27, gives

$$\dot{\dot{L}}_3 = \frac{X_p \ddot{X}_p + \dot{X}_p^2 + Y_p \ddot{Y}_p + \dot{Y}_p^2 + Z_p \ddot{Z}_p + \dot{Z}_p^2 - \dot{L}_3^2}{L_3} \quad (6-28)$$

for the acceleration.

Assuming that the platform motion (  $\mathbf{U}$ ,  $\dot{\mathbf{U}}$  and  $\ddot{\mathbf{U}}$  ) is known, comparing the results calculated from equation 6-24 with equation 6-27, and equation 6-25 with equation 6-28 for each leg, that is,

$$[\dot{\mathbf{d}}]_{r,i} = \dot{L}_3 \quad (6-29)$$

and

$$[\ddot{\mathbf{d}}]_{r,i} = \dot{\dot{L}}_3 \quad (6-30)$$

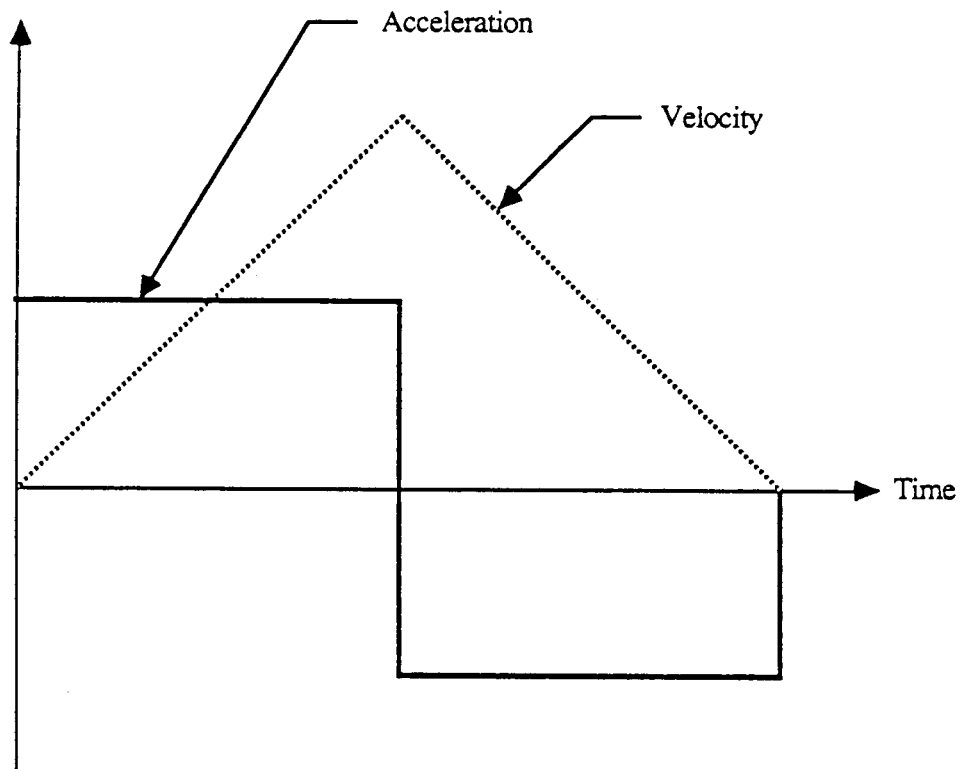
where  $r=1, 2, \dots, 6$  is the leg number, verified the G- and H-functions derived for the model. This is because  $[\mathbf{G}_u^d]$  and  $[\mathbf{H}_{uu}^d]$  are derived using  $[\mathbf{G}_\phi^p]$ ,  $[\mathbf{H}_{\phi\phi}^p]$ ,  $[\mathbf{G}_u^p]$ ,  $[\mathbf{H}_{uu}^p]$  and all the intermediate G- and H-functions. If  $[\mathbf{G}_u^d]$  and  $[\mathbf{H}_{uu}^d]$  are checked,

all the other G- and H-functions are checked too. This concludes the verification for the G- and H-functions.

## 6-2 Simulation of the Dynamic Docking Test System(DDTS)

This section discusses several computer simulations for the DDTS based on the model established in Chapter 5. The simulations are coded in FORTRAN language employing a VAX/VMS 11/750 computer. A complete computer listing of the program written for one of the simulations is listed in Appendix B. The computer listing is for the case when the platform is translating and rotating about the X\* axis using a class p=3, 3-4-5 polynomial curve, which will be discussed very shortly. To use the program for any other motion, one only needs to modify the subroutine "NEXT\_MOTION" and the respective parameters in the calling statement. The input data corresponding to the subroutine "READ\_DATA" for the program listed in Appendix B is listed in Appendix C. The main purpose of these simulations is to investigate the relative contribution to the overall generalized forces, required at each actuator, of the acceleration related (ie.,  $\begin{bmatrix} \ddot{I}_{dd} \end{bmatrix}$ ) and the velocity related (ie.,  $\begin{bmatrix} \dot{P}_{dd} \end{bmatrix}$ ) terms.

Four different motion specifications for the platform are adopted from Matthew and Tesar [29]. The first motion is the class p=2, constant acceleration curve as shown in Fig. 6-1. The subroutine "NEXT\_MOTION" for this motion is listed in Appendix D. The second motion is the class p=3, 3-4-5 polynomial curve as in Fig. 6-2. The last two motions both belong to class p=4, one of them is the 4-5-6-7 polynomial curve and the other one is a 3<sup>rd</sup> derivative trapezoidal curve, as illustrated in Fig. 6-3 and 6-4, respectively. Similarly, the subroutines corresponding to these two motions are listed in Appendix E and F, respectively. The use of class p=2, 3, or 4 in this context indicates that a jump or discontinuity occurs in the p<sup>th</sup> derivative of the motion function. The expressions for the displacement, velocity and acceleration corresponding to each of the motion curves are also listed along with the figures. The maximum displacement is denoted by  $\bar{y}$  and the total time taken is denoted by  $\bar{t}$ . Note that the 3<sup>rd</sup> derivative

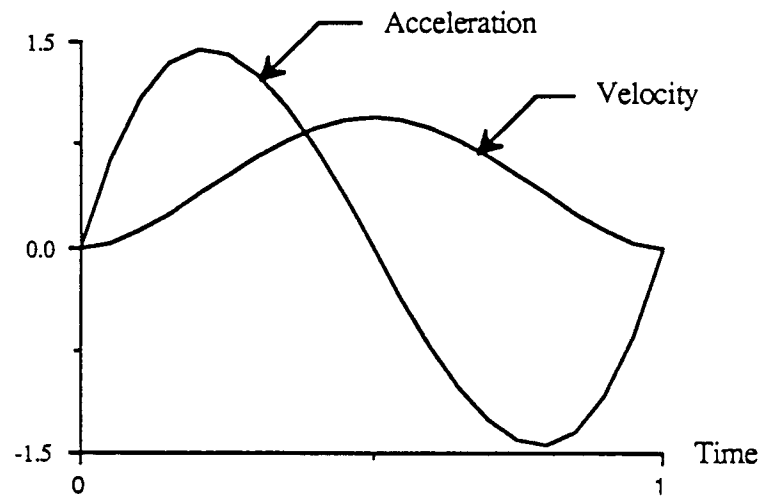


$$y = \frac{2\bar{y}}{\bar{t}^2} \left[ t^2 - 2 \left\langle t - \frac{\bar{t}}{2} \right\rangle^2 \right]$$

$$y' = \frac{4\bar{y}}{\bar{t}^2} \left[ t - 2 \left\langle t - \frac{\bar{t}}{2} \right\rangle \right]$$

$$y'' = \frac{4\bar{y}}{\bar{t}^2} \left[ 1 - 2 \left\langle t - \frac{\bar{t}}{2} \right\rangle^0 \right]$$

Fig. 6-1 Class  $p=2$ , Constant Acceleration

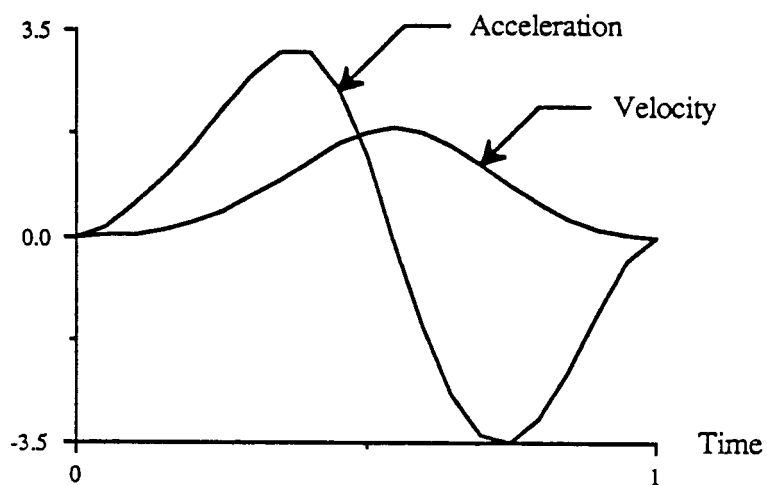


$$y = \bar{y} \left[ 10 \left( \frac{t}{t} \right)^3 - 15 \left( \frac{t}{t} \right)^4 + 6 \left( \frac{t}{t} \right)^5 \right]$$

$$\dot{y} = \frac{\bar{y}}{t} \left[ 30 \left( \frac{t}{t} \right)^2 - 60 \left( \frac{t}{t} \right)^3 + 30 \left( \frac{t}{t} \right)^4 \right]$$

$$\ddot{y} = \frac{\bar{y}}{t^2} \left[ 60 \left( \frac{t}{t} \right) - 180 \left( \frac{t}{t} \right)^2 + 120 \left( \frac{t}{t} \right)^3 \right]$$

Fig. 6-2 Class p=3, 3-4-5 Polynomial

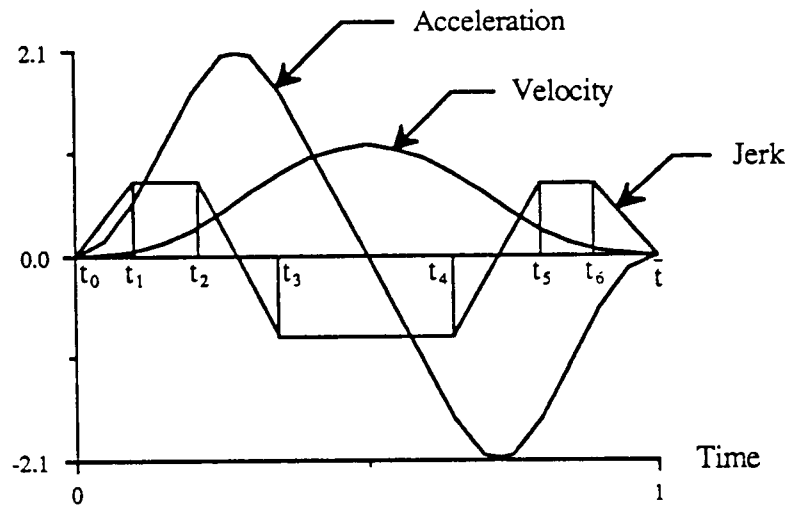


$$y = \bar{y} \left[ 35 \left( \frac{t}{t} \right)^4 - 84 \left( \frac{t}{t} \right)^5 + 70 \left( \frac{t}{t} \right)^6 - 20 \left( \frac{t}{t} \right)^7 \right]$$

$$y' = \frac{\bar{y}}{t} \left[ 140 \left( \frac{t}{t} \right)^3 - 420 \left( \frac{t}{t} \right)^4 + 420 \left( \frac{t}{t} \right)^5 - 140 \left( \frac{t}{t} \right)^6 \right]$$

$$y'' = \frac{\bar{y}}{t^2} \left[ 420 \left( \frac{t}{t} \right)^2 - 1680 \left( \frac{t}{t} \right)^3 + 2100 \left( \frac{t}{t} \right)^4 - 840 \left( \frac{t}{t} \right)^5 \right]$$

Fig. 6-3 Class  $p=4$ , 4-5-6-7 Polynomial



$$\begin{aligned}
 y^{(4 \cdot i)} = & \frac{A_1}{i!} \left[ \frac{\langle t-t_0 \rangle^i - \langle t-t_1 \rangle^i}{t_1-t_0} - \frac{\langle t-t_2 \rangle^i - \langle t-t_3 \rangle^i}{t_3-t_2} \right] \\
 & + \frac{A_2}{i!} \left[ \frac{\langle t-t_2 \rangle^i - \langle t-t_3 \rangle^i}{t_3-t_2} - \frac{\langle t-t_4 \rangle^i - \langle t-t_5 \rangle^i}{t_5-t_4} \right] \\
 & + \frac{A_3}{i!} \left[ \frac{\langle t-t_4 \rangle^i - \langle t-t_5 \rangle^i}{t_5-t_4} - \frac{\langle t-t_6 \rangle^i - \langle t-\bar{t} \rangle^i}{\bar{t}-t_6} \right]
 \end{aligned}$$

$$y:i=4; \quad y':i=3; \quad y'':i=2$$

Fig. 6-4 Class p=4, 3<sup>rd</sup> Derivative Trapezoidal

trapezoidal curve involves three constants A1, A2 and A3. They can be determined by using the expression

$$\begin{bmatrix} A_1 \\ A_2 \\ A_3 \end{bmatrix} = [t_{ij}] \begin{bmatrix} 2\bar{y}'' - 2C_1 \\ 6\bar{y}' - 6(C_1\bar{t} + C_2) \\ 24\bar{y} - 24\left(\frac{C_1}{2}\bar{t}^2 + C_2\bar{t} + C_3\right) \end{bmatrix} \quad (6-31)$$

and

$$t_{ij} = \frac{(\bar{t} - t_{2j-2})^{i+1} - (\bar{t} - t_{2j-1})^{i+1}}{t_{2j-1} - t_{2j-2}} - \frac{(\bar{t} - t_{2j})^{i+1} - (\bar{t} - t_{2j+1})^{i+1}}{t_{2j+1} - t_{2j}} \quad (6-32)$$

where  $C_1, C_2, C_3$  are the integration constants resulting from the initial conditions. Detailed derivation of equations 6-31 and 6-32 is given in Matthew and Tesar [29]. The subroutine used in the simulation to determine the constants A1, A2 and A3 is listed in Appendix G.

Besides employing different classes of motion specifications, two arbitrarily chosen platform trajectories are investigated. First, the platform moves vertically along the  $X^*$  axis from 3m to 4m, while simultaneously rotating about the same axis from 0 to -60 degrees, as shown in Fig. 6-5. In the second trajectory, the platform moves horizontally along the  $Z^*$  axis from 0m to 1m while  $X^*$  remains constant at 3m. As before, the platform rotates about the  $X^*$  axis from 0 to -60 degrees. This trajectory is illustrated in Fig. 6-6. For the second trajectory, only the class  $p=3, 3-4-5$  polynomial motion is investigated. In addition, a different investigation is also conducted to study the relative contribution of the platform and leg mass to the effective inertia matrix and the inertia power array. This can be achieved by first assuming that only the platform has mass while the legs are massless, then assuming that the six legs have mass while the platform is massless. This investigation is conducted using the trajectory in Fig. 6-6 and the

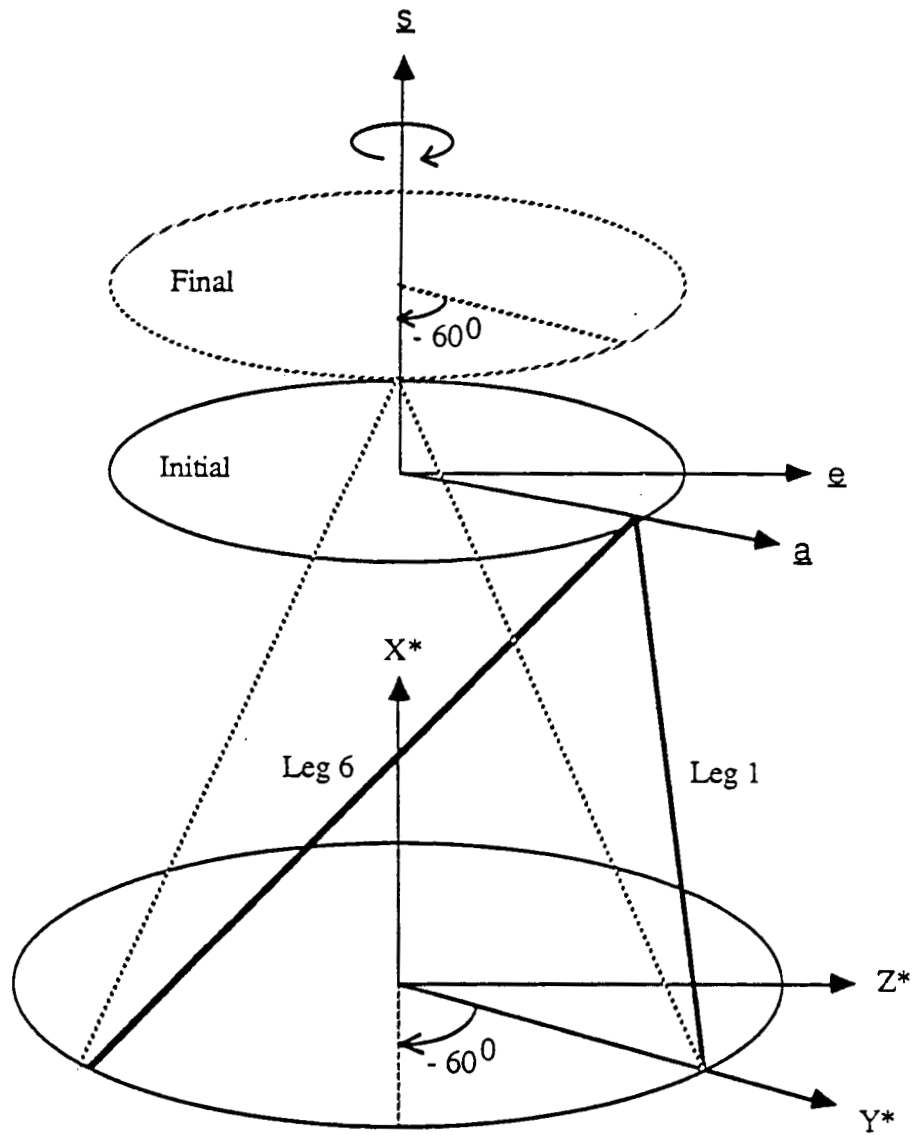


Fig. 6-5 Translating and Rotating about  $X^*$  Axis

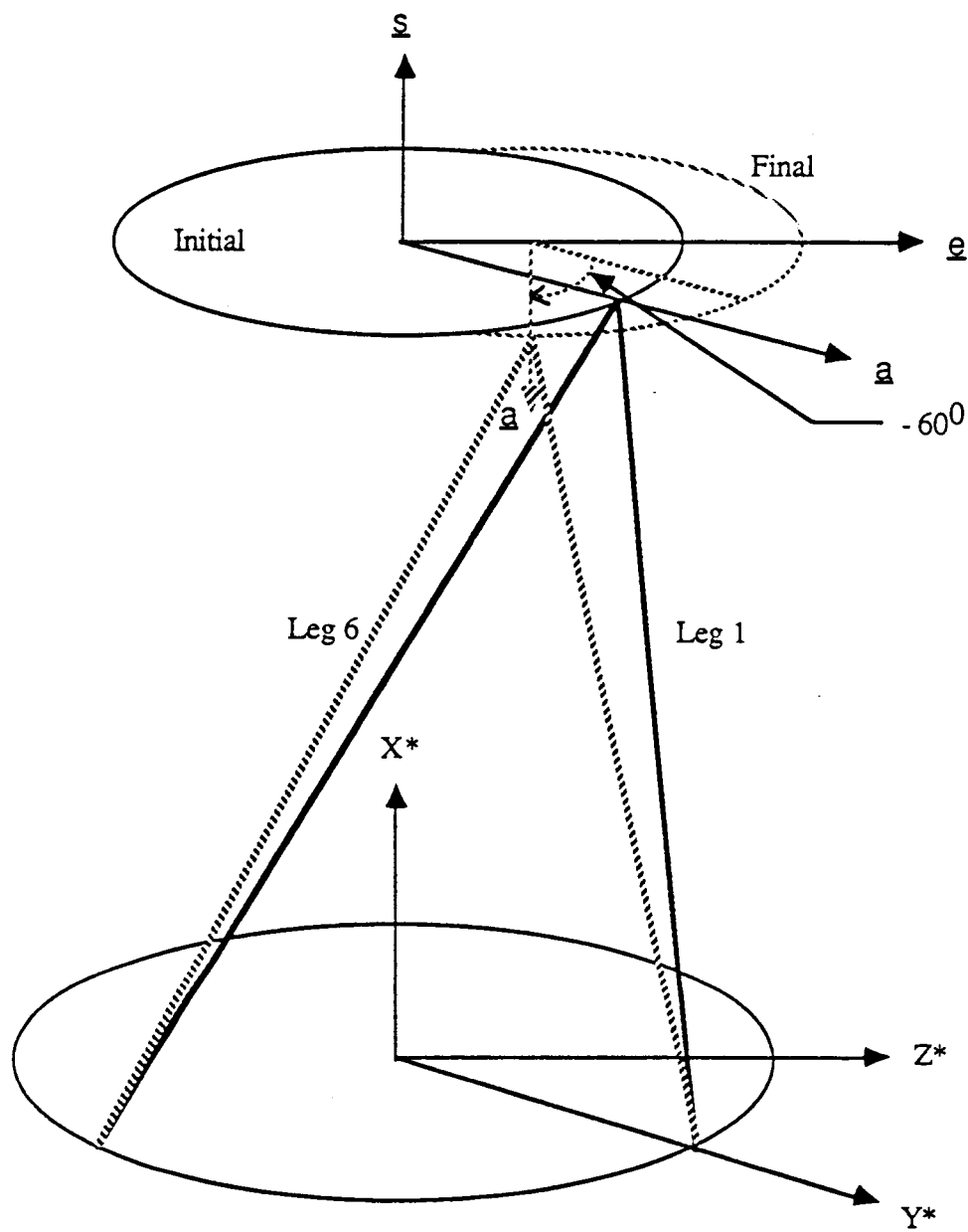


Fig. 6-6 Translating Along  $Z^*$  Axis  
Rotating About  $X^*$  Axis

class  $p=3$ , 3-4-5 polynomial, since it is felt that this trajectory provides a more general study due to its non-symmetric nature. The total time taken for each motion is 2sec, with an interval of 0.1sec for every time step. The dimensions of the base, platform, legs and all other components used in the simulations are listed in Table 6-1. Note that the numbers adopted here are all estimated values based on a photograph of the DDTS since the actual numbers were unavailable. The mass of the six legs and the platform are intentionally slightly overestimated to accommodate the likely application of the mechanism for large space vehicle docking purposes.

The resulting actuator motions ( ie.,  $\underline{d}$ ,  $\dot{\underline{d}}$  and  $\ddot{\underline{d}}$  ) and the required generalized forces at each actuator, along with the contribution from the velocity and the acceleration related terms, are plotted in Fig. 6-7 through Fig. 6-29. The notation used to denote the figures is a set of five alphanumericics. The code starts with an uppercase letter denoting which platform trajectory is being employed. "D" is for the trajectory that translates and rotates about the  $X^*$  axis. "E" refers to the trajectory that translates along the  $Z^*$  axis and rotates about the  $X^*$  axis. The following digit indicates which type of motion specification is used. The digit "4" is for the class  $p=2$ , constant acceleration, "5" is for the class  $p=3$ , 3-4-5 polynomial, "6" is the class  $p=4$ , 4-5-6-7 polynomial, and finally, "7" refers to the class  $p=4$ , 3<sup>rd</sup> derivative trapezoidal curve. The last digit denotes the total time allowed, which in this work is always 2 for two seconds. The last lowercase letter refers to the leg number, from "a" for leg 1 to "f" for leg 6. The two zeros in the code serve no practical purpose. For the first trajectory (ie., translating and rotating along/about the  $X^*$  axis), the required generalized force curves of each leg are plotted following the plot for the position, velocity and acceleration of the actuator for that particular leg.

### 6-3 Conclusions

Comparing the plots for the platform motion ( ie.,  $\underline{U}$ ,  $\dot{\underline{U}}$  and  $\ddot{\underline{U}}$  ) with the plots for each actuator's motion (ie.,  $\underline{d}$ ,  $\dot{\underline{d}}$  and  $\ddot{\underline{d}}$  ) does not show a direct relationship between them. This implies that in equations 6-24 and 6-25,

Radius of the base =  $B = 3$  m

Radius of the platform =  $R = 2$  m

Radius of the Hooke joint =  $r = 0.06$  m  
(modeled as cylinder)

Length of the piston cylinder =  $L_c = 3$  m

Length of the piston rod =  $L_r = 3$  m

Mass of the Hooke joint =  $M_{12} = 10.0$  kg

Mass of the piston cylinder =  $M_{23} = 200$  kg

Mass of the piston rod =  $M_{34} = 175$  kg

Mass of the platform =  $M_{pl} = 2250$  kg

**Table 6-1 Estimated Dimensions and Mass of the  
Generalized Stewart Platform**

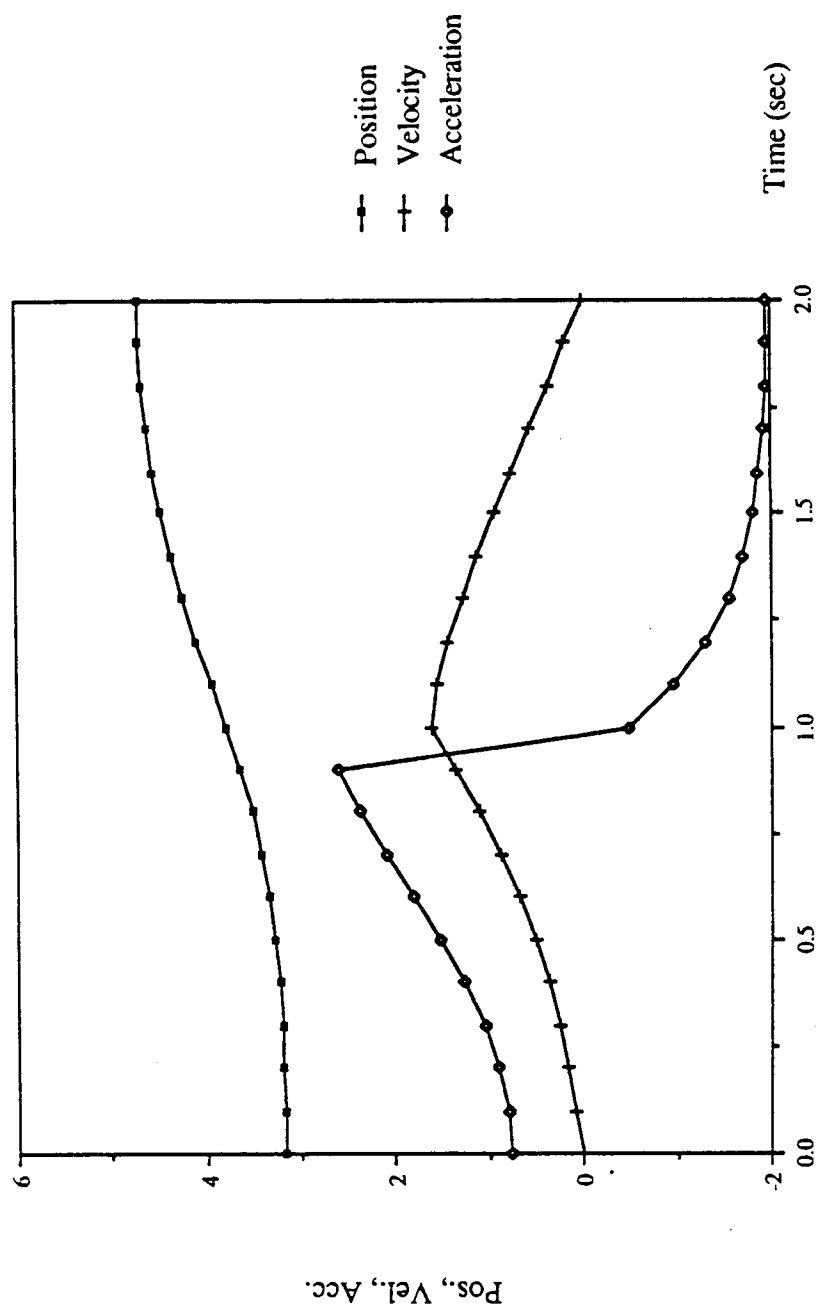


Fig. 6-7 Actuator Motion D4002a

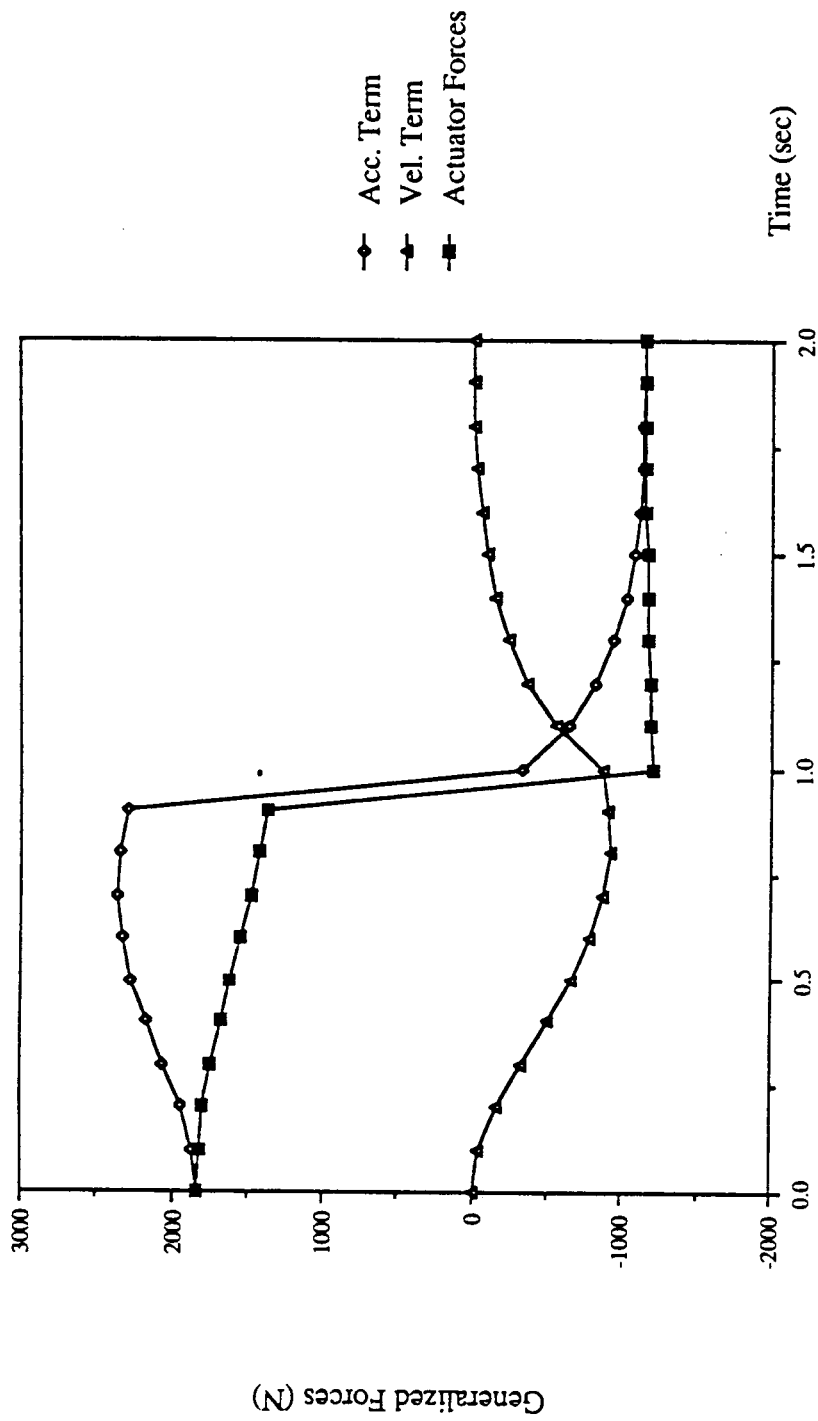


Fig. 6-8 Actuator Forces D4002a

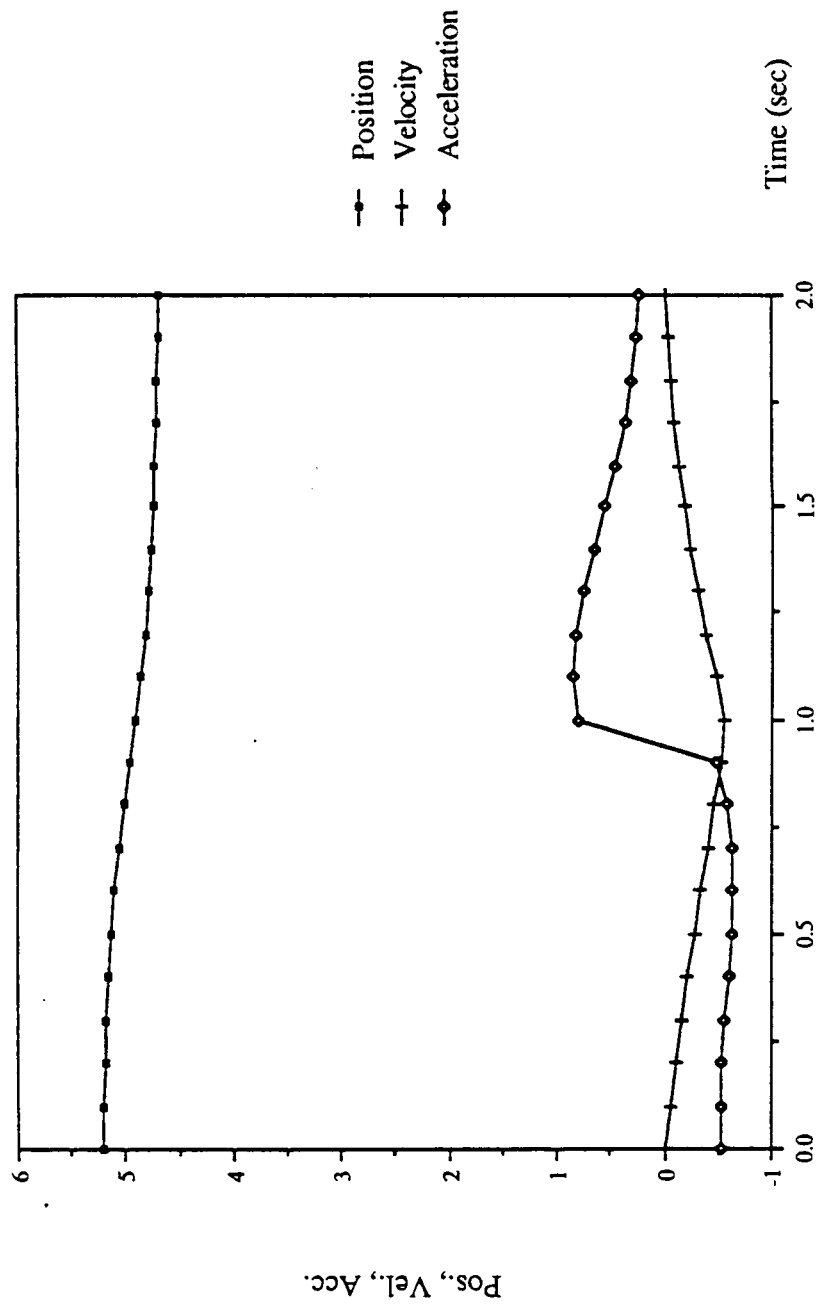


Fig. 6-9 Actuator Motion D4002b

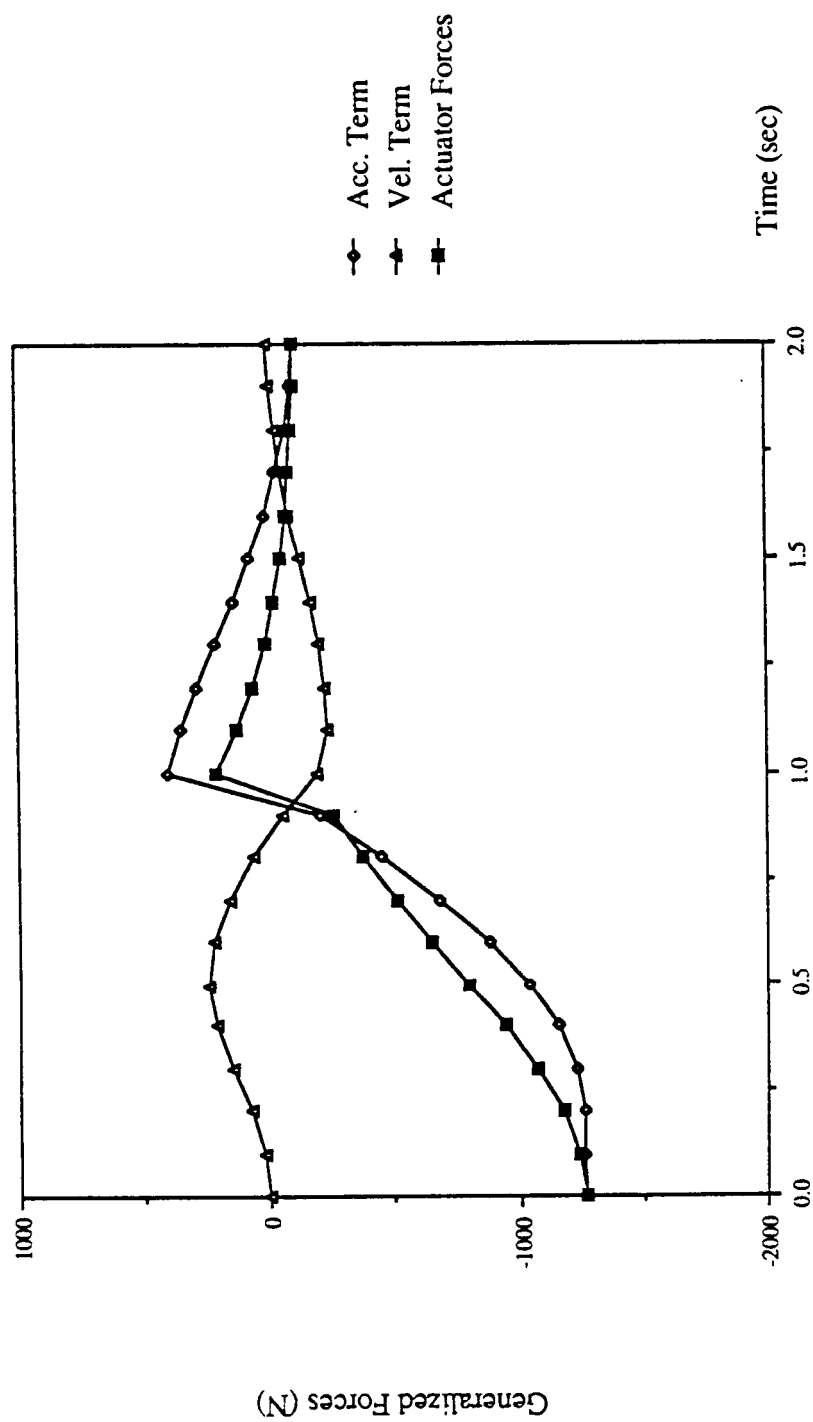


Fig. 6-10 Actuator Forces D4002b

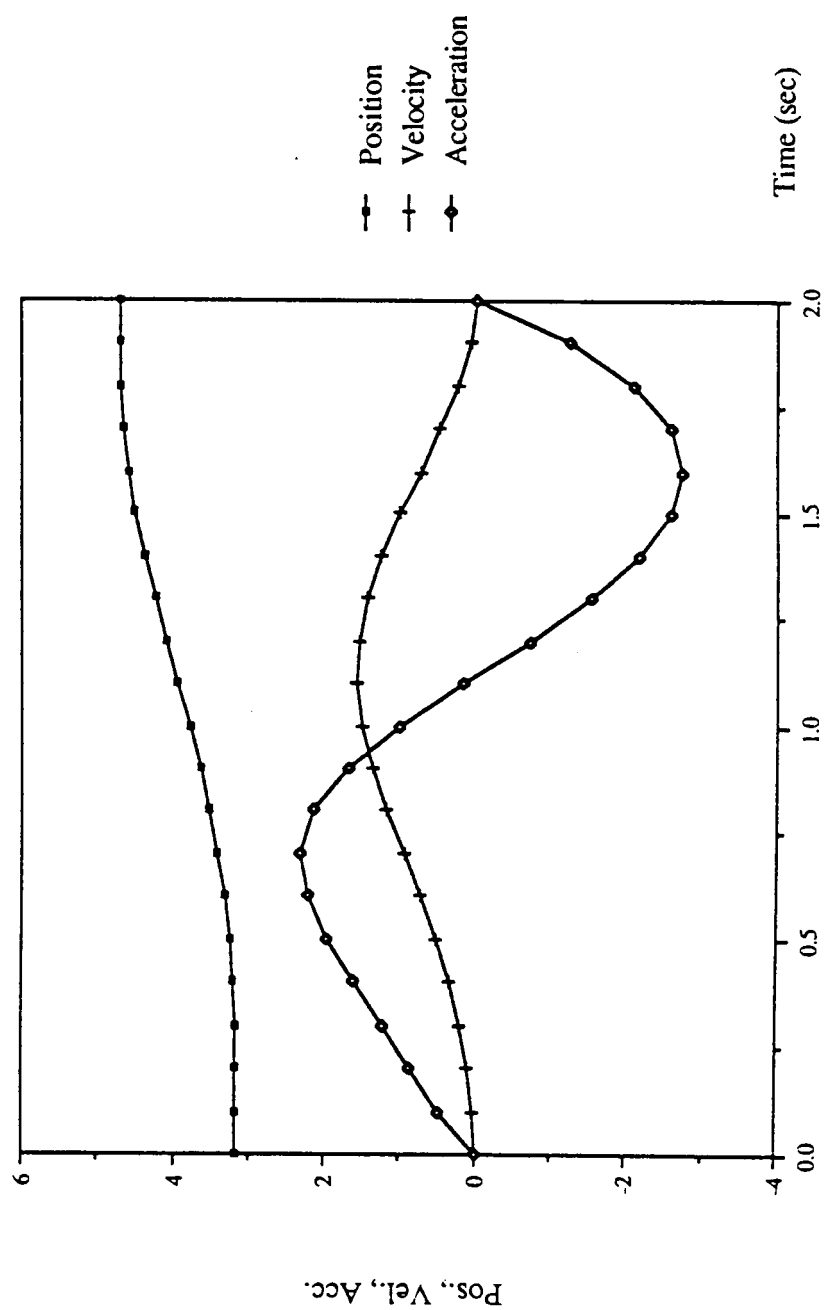


Fig. 6-11 Actuator Motion D5002a

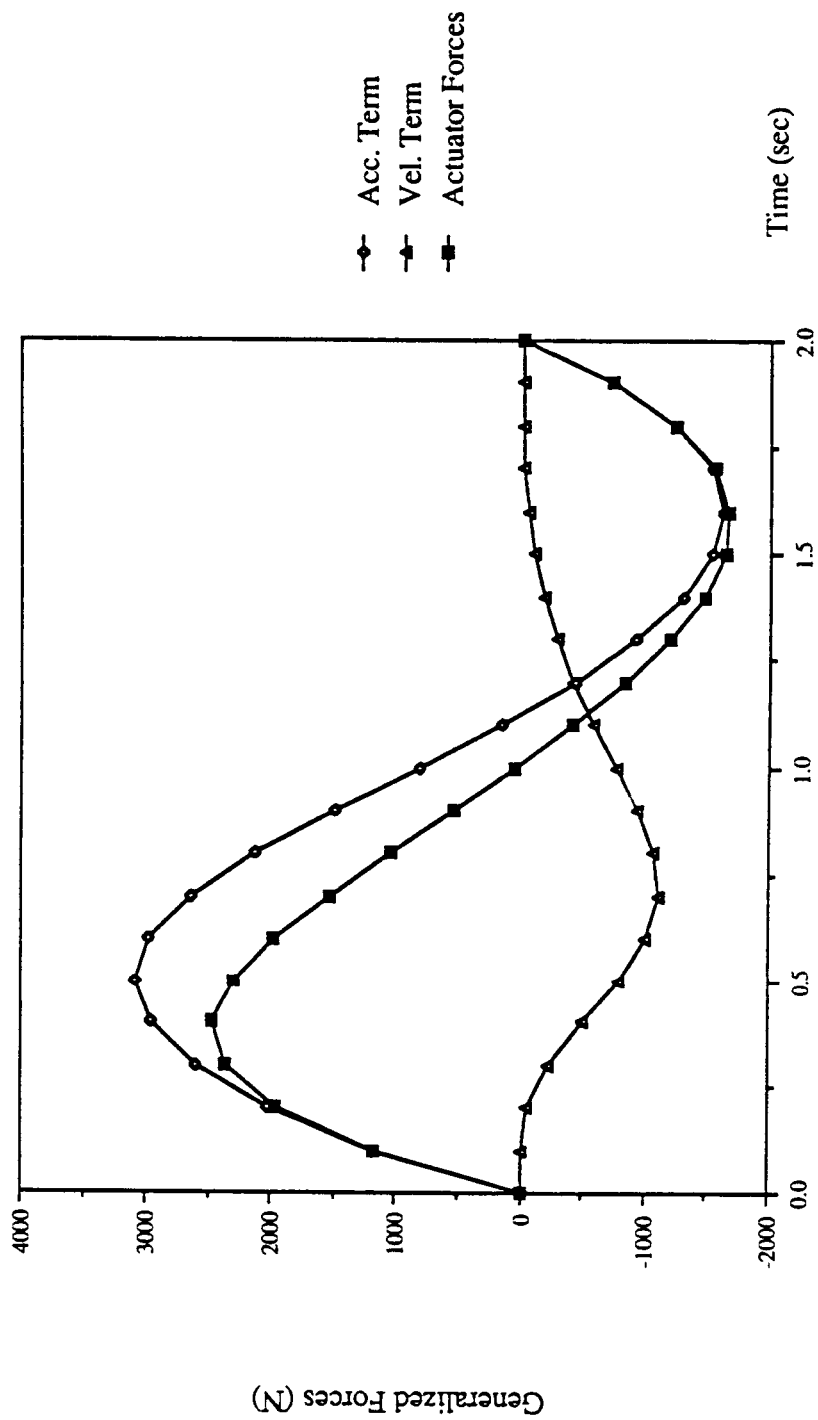


Fig. 6-12 Actuator Forces D5002a

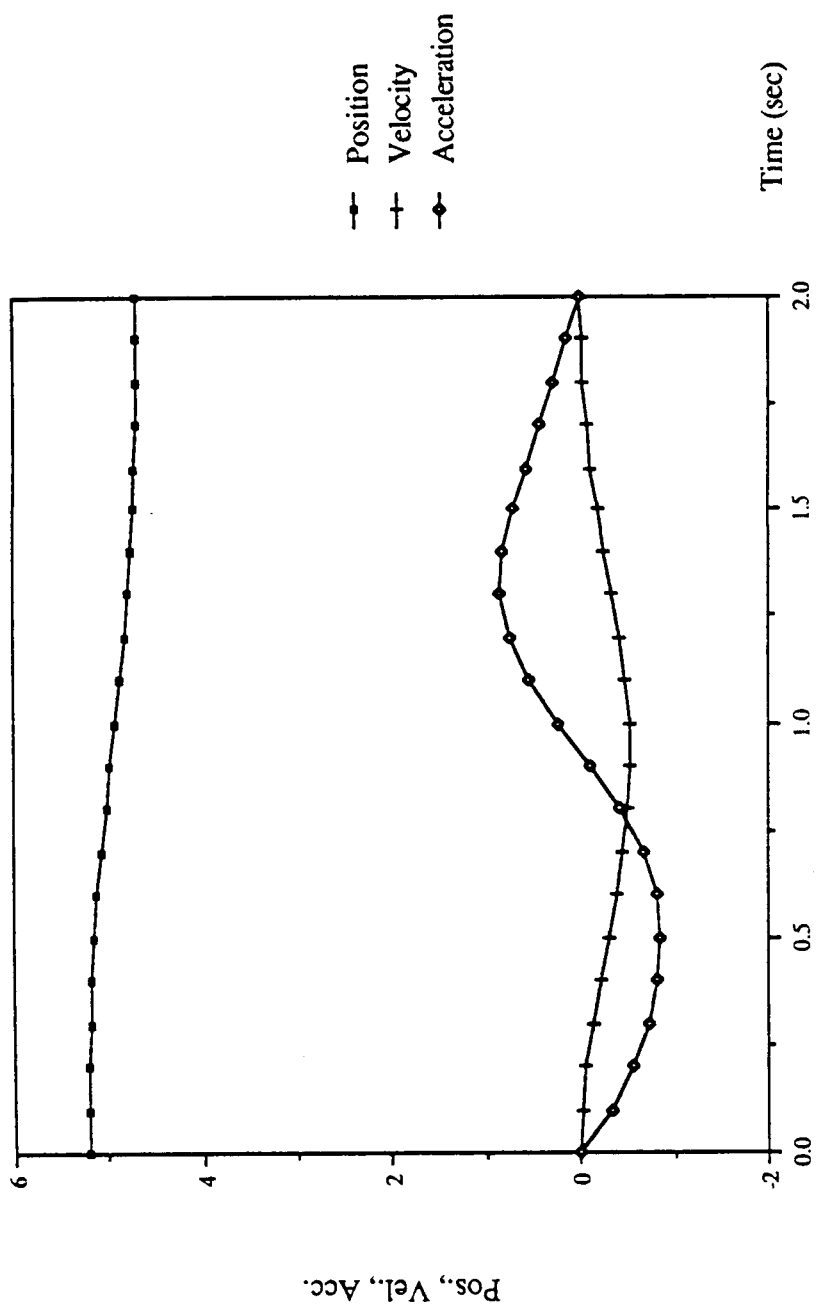


Fig. 6-13 Actuator Motion D5002b

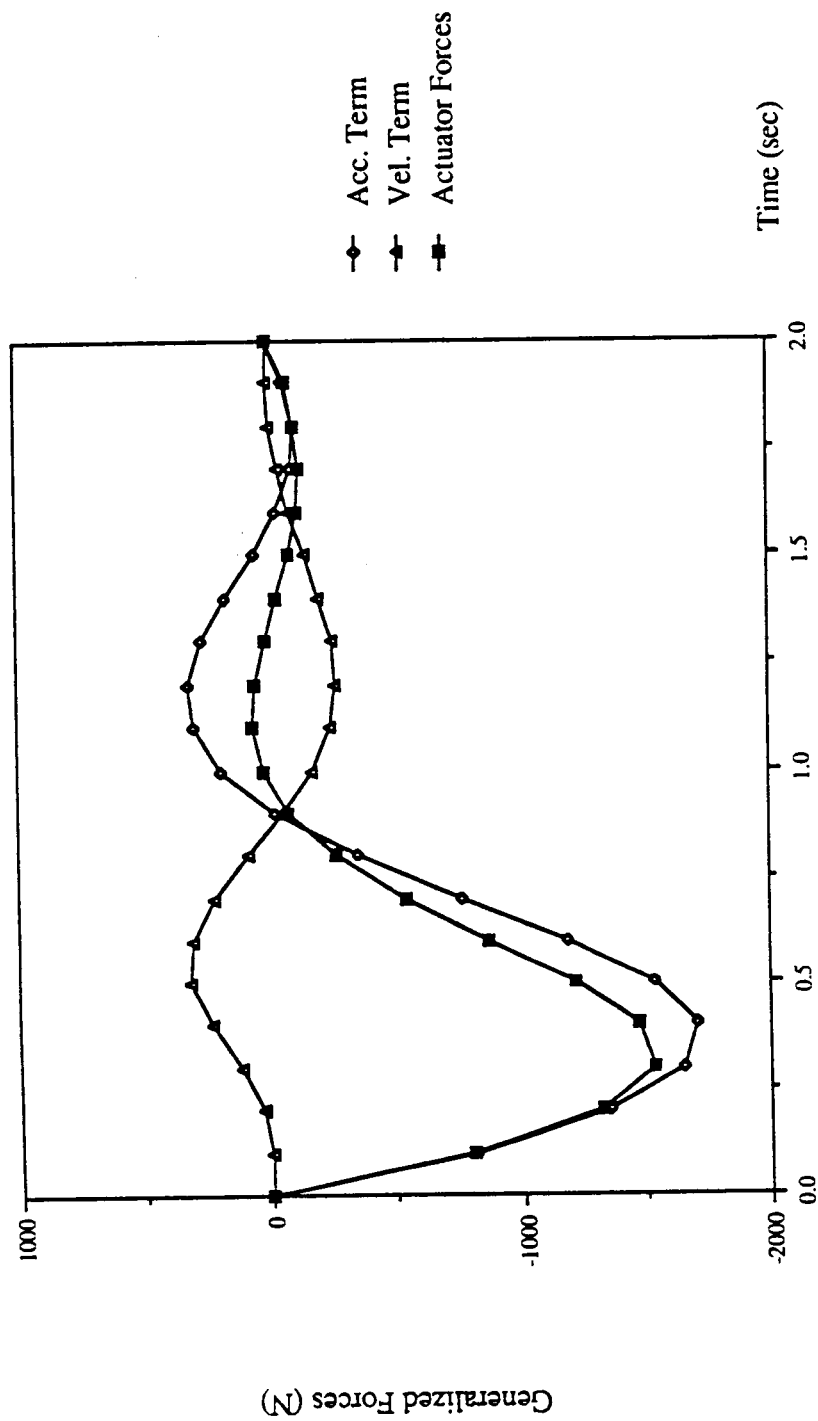


Fig. 6-14 Actuator Forces D5002b

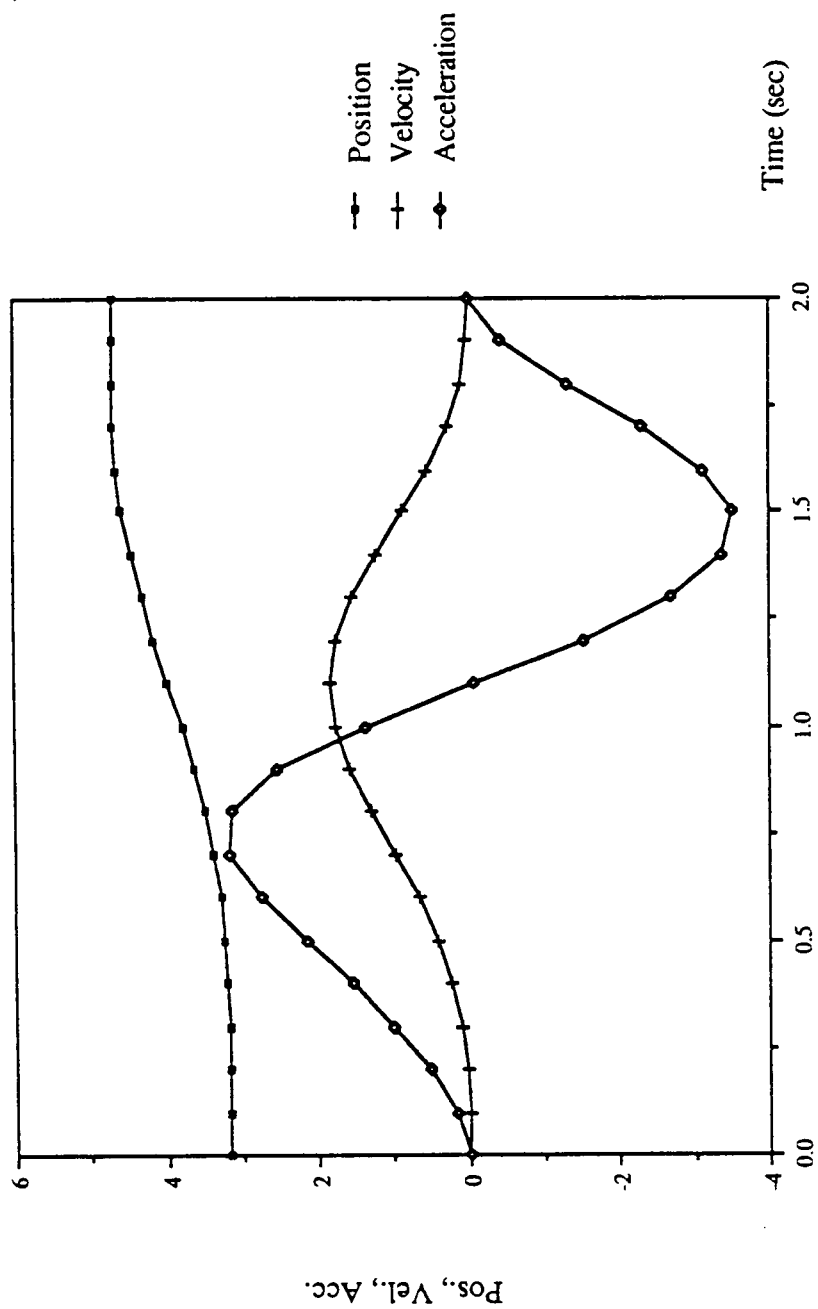


Fig. 6-15 Actuator Motion D6002a

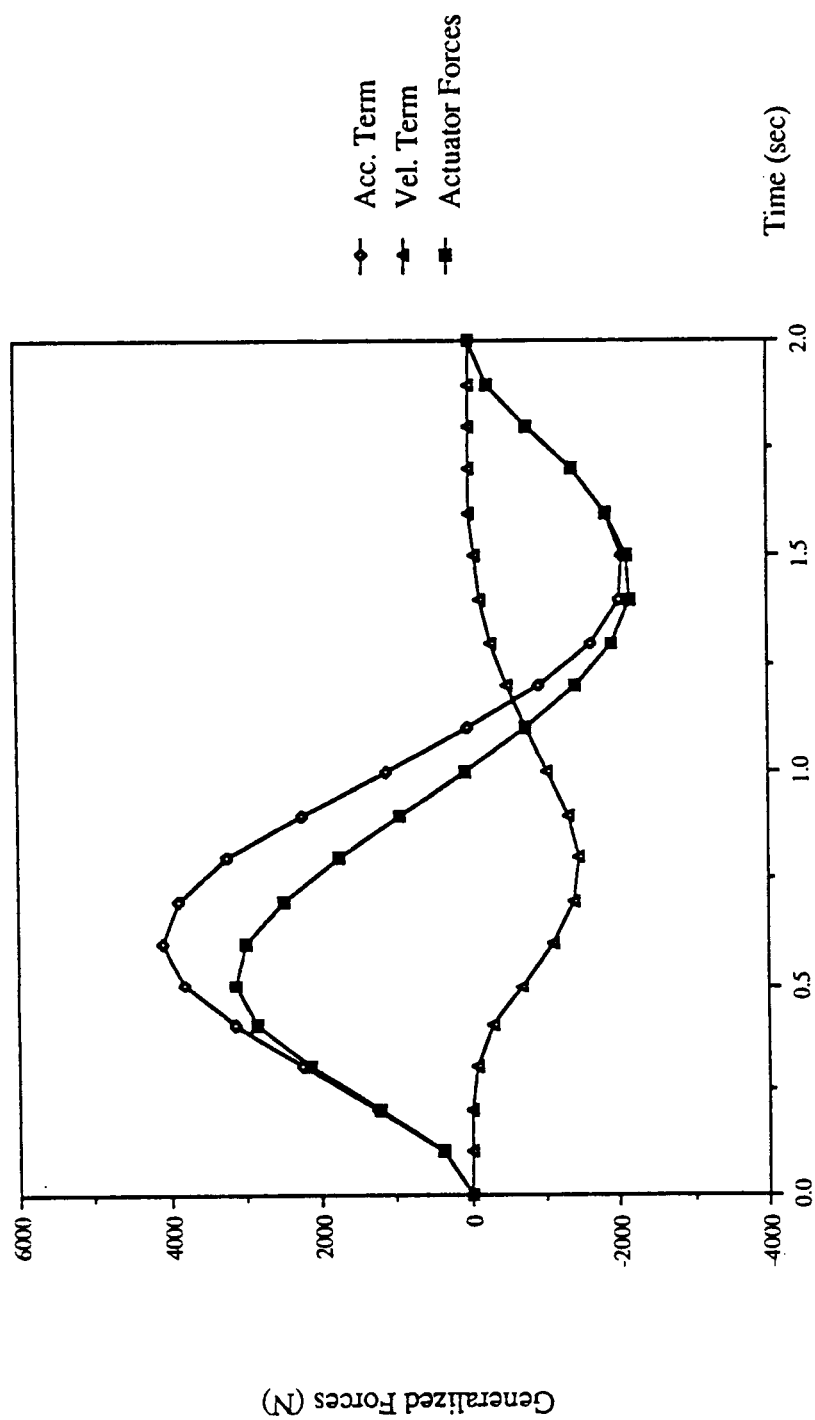
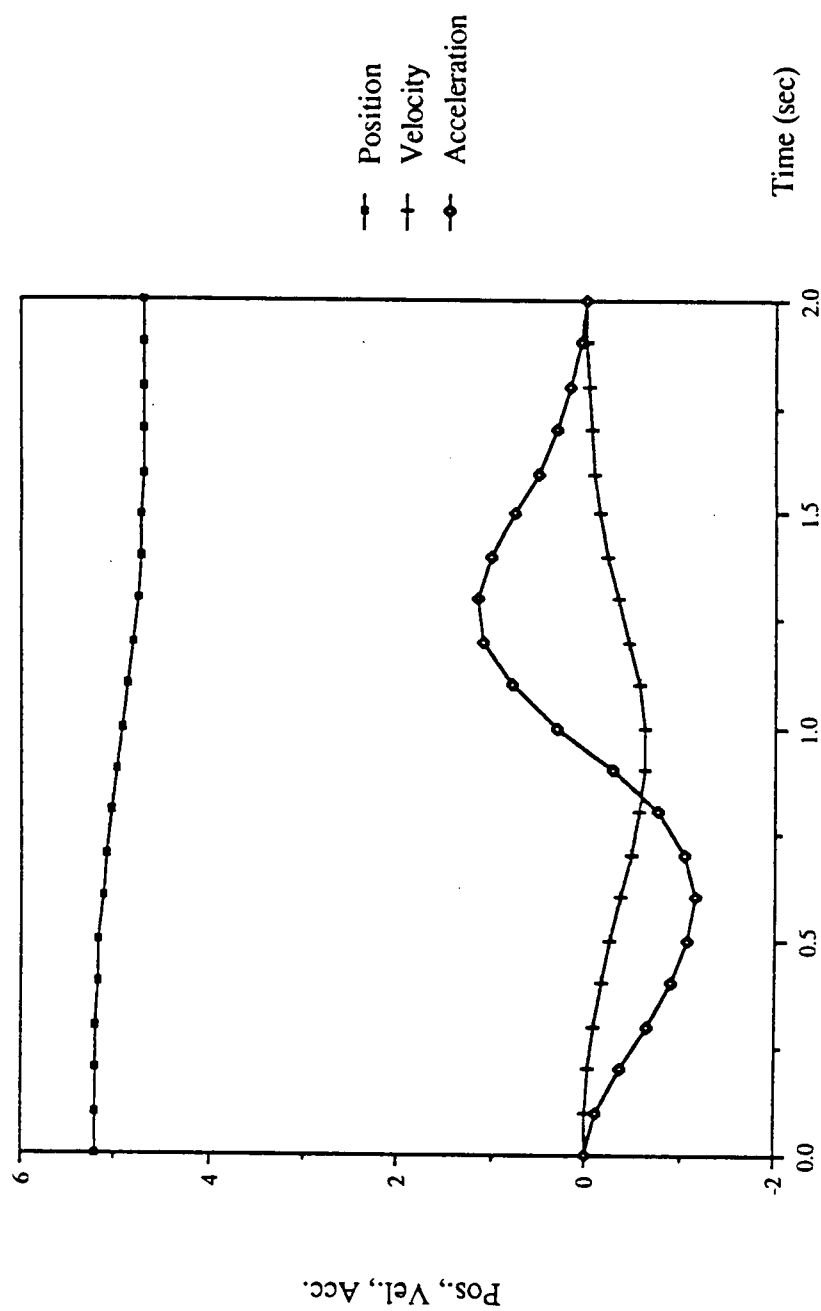


Fig. 6-16 Actuator Forces D6002a

**Fig. 6-17 Actuator Motion D6002b**

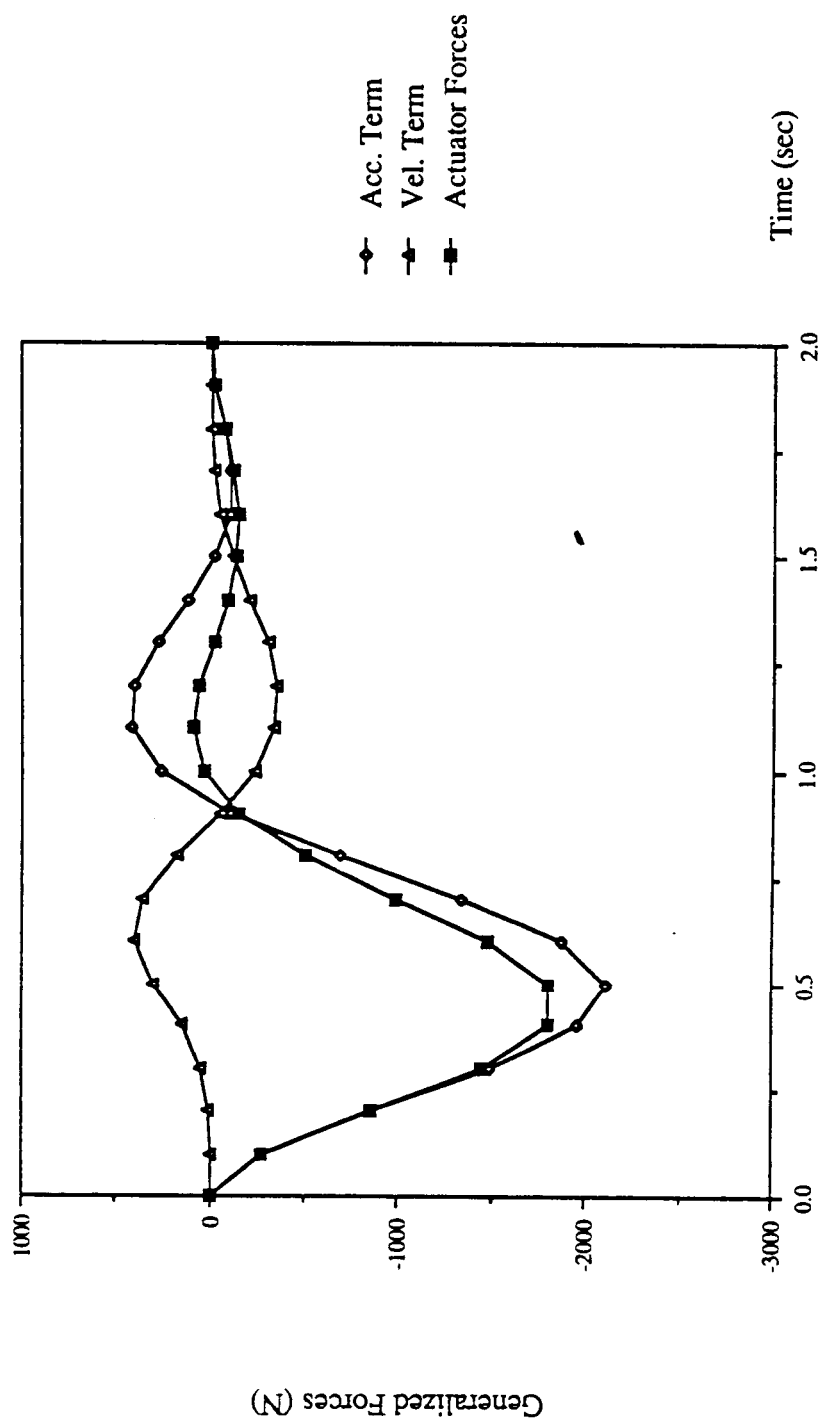


Fig. 6-18 Actuator Forces D6002b

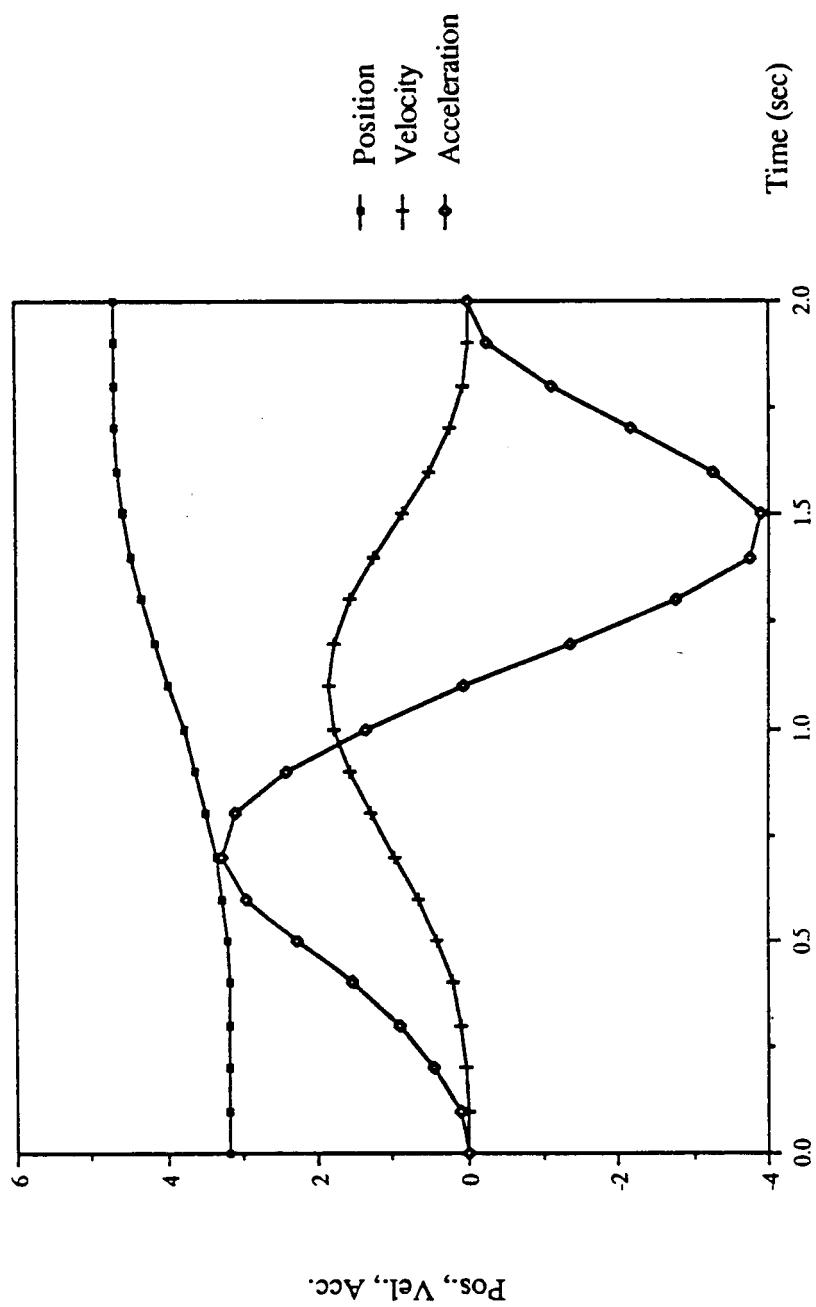


Fig. 6-19 Actuator Motion D7002a

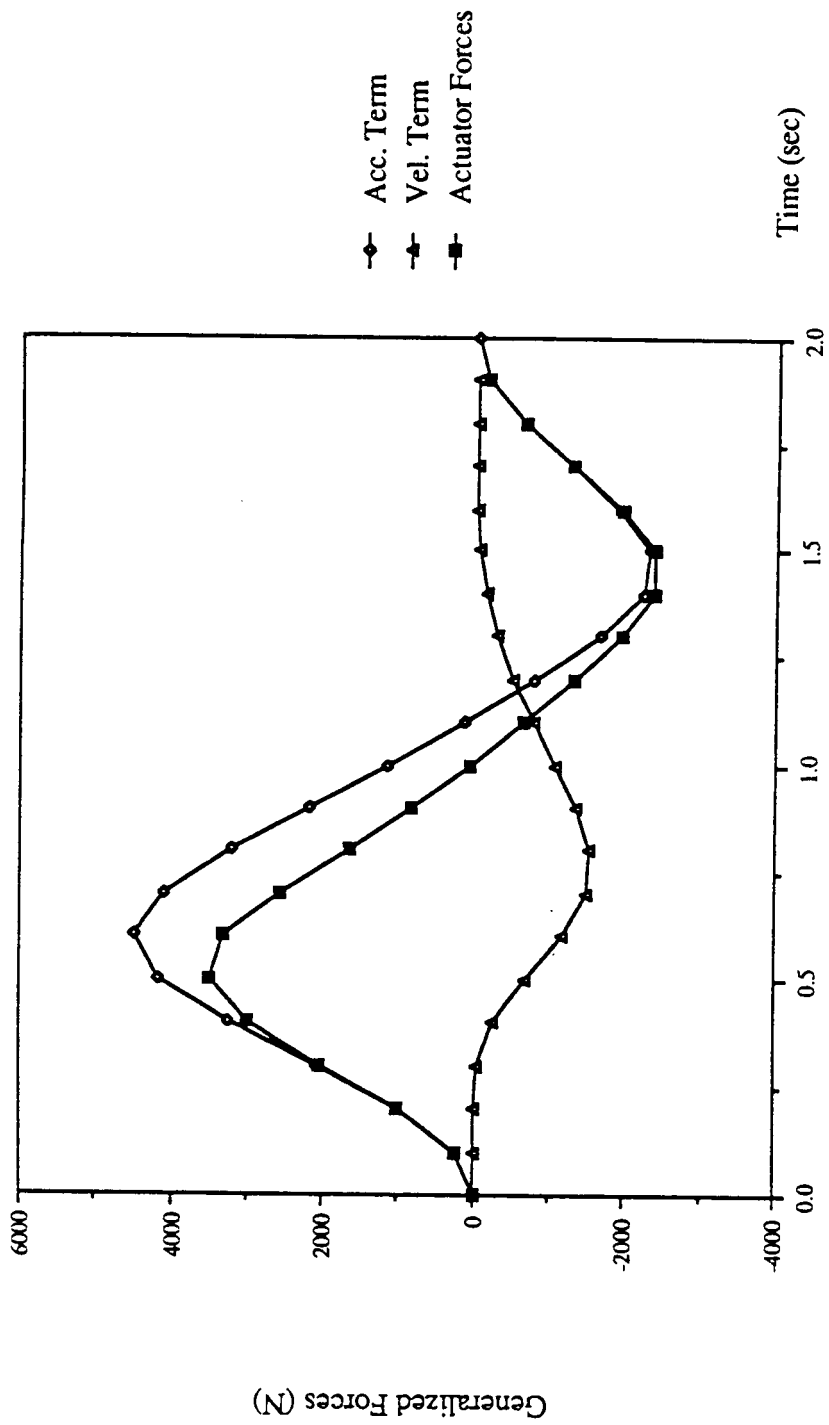


Fig. 6-20 Actuator Forces D7002a

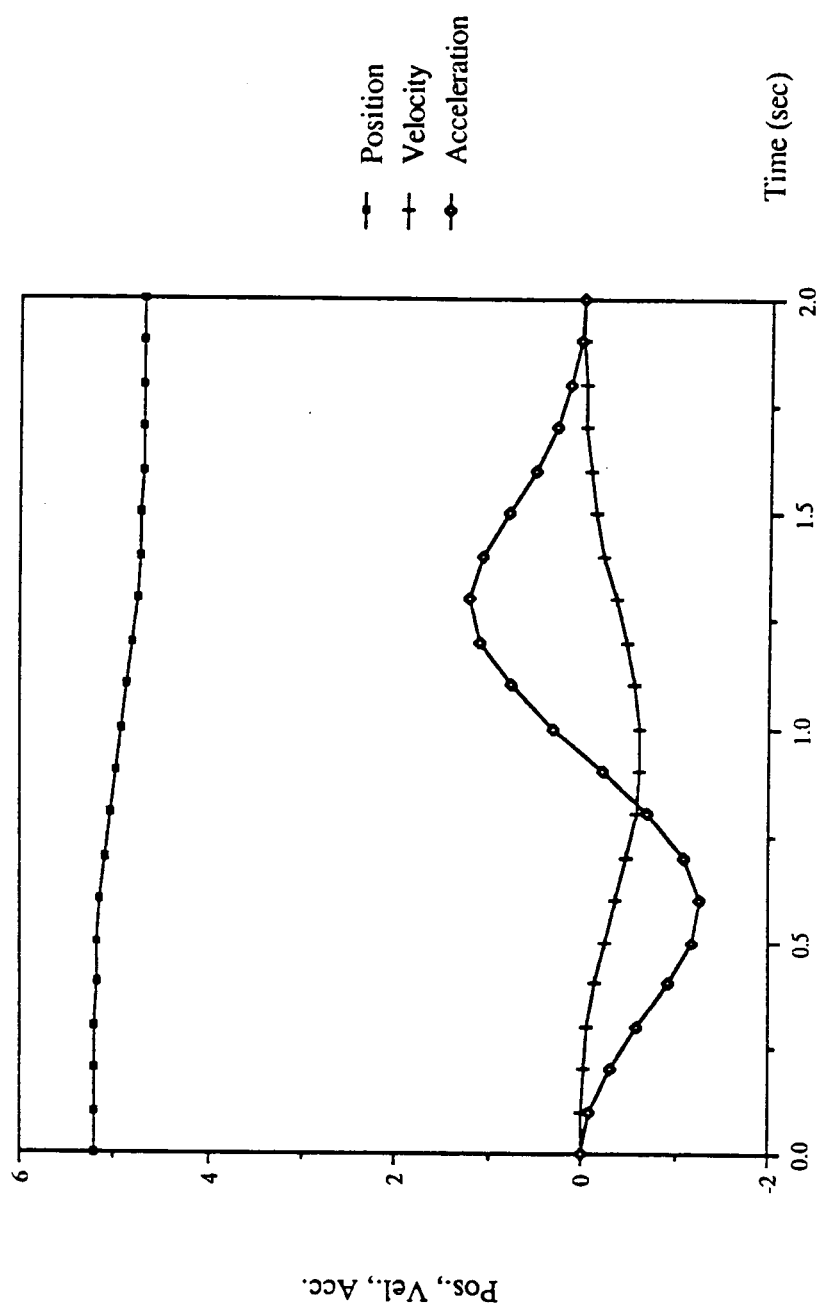


Fig. 6-21 Actuator Motion D7002b

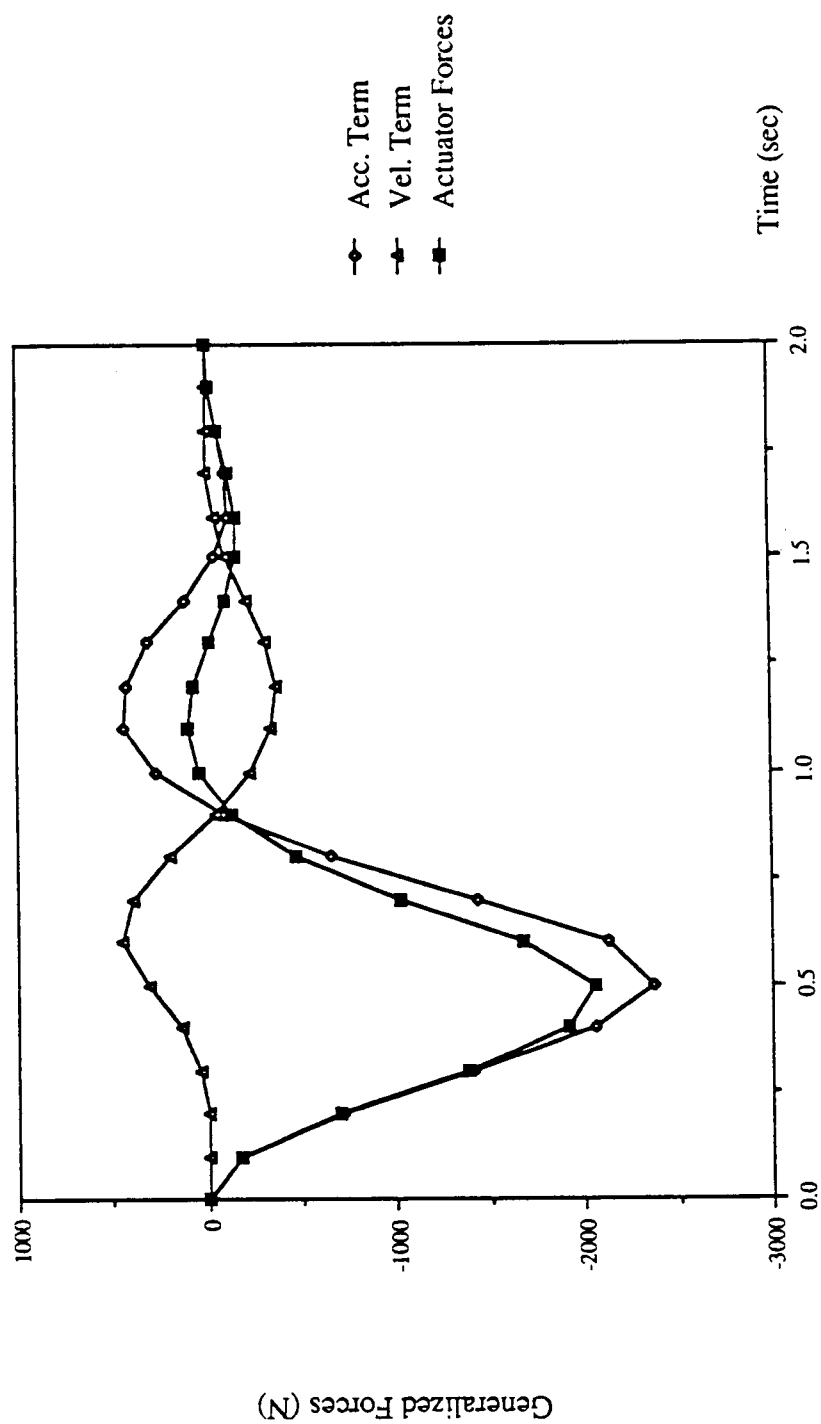


Fig. 6-22 Actuator Forces D7002b

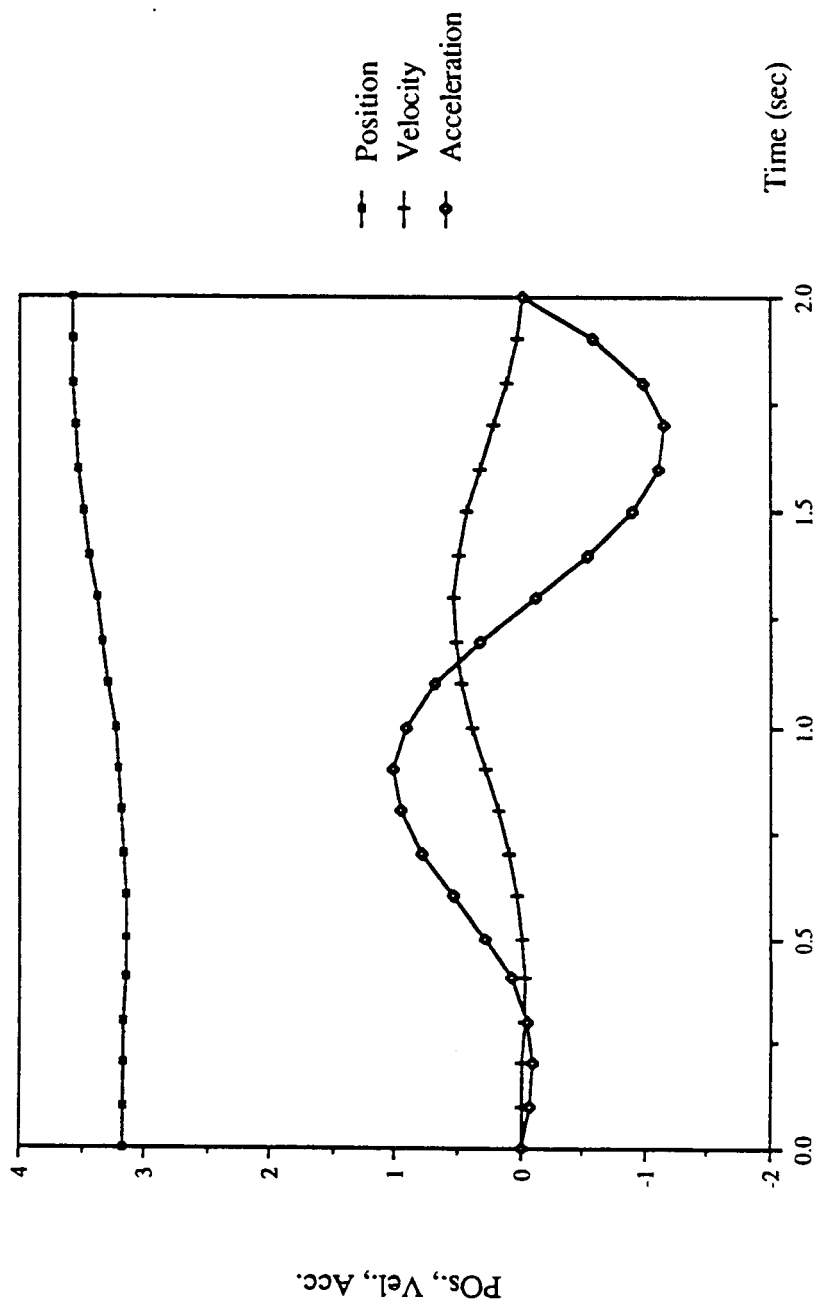


Fig. 6-23 Actuator Motion E5002a

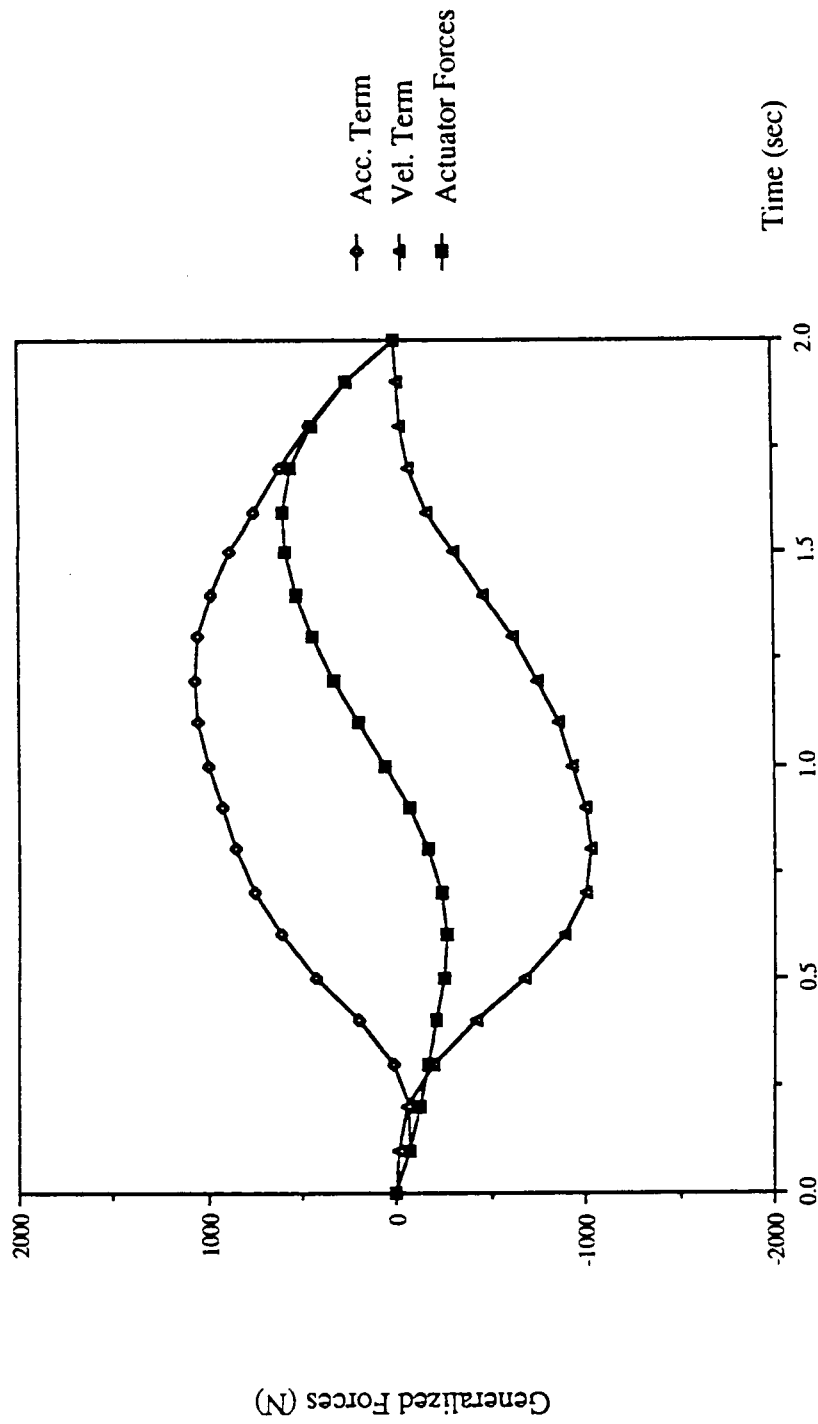


Fig. 6-24 Actuator Forces E5002a

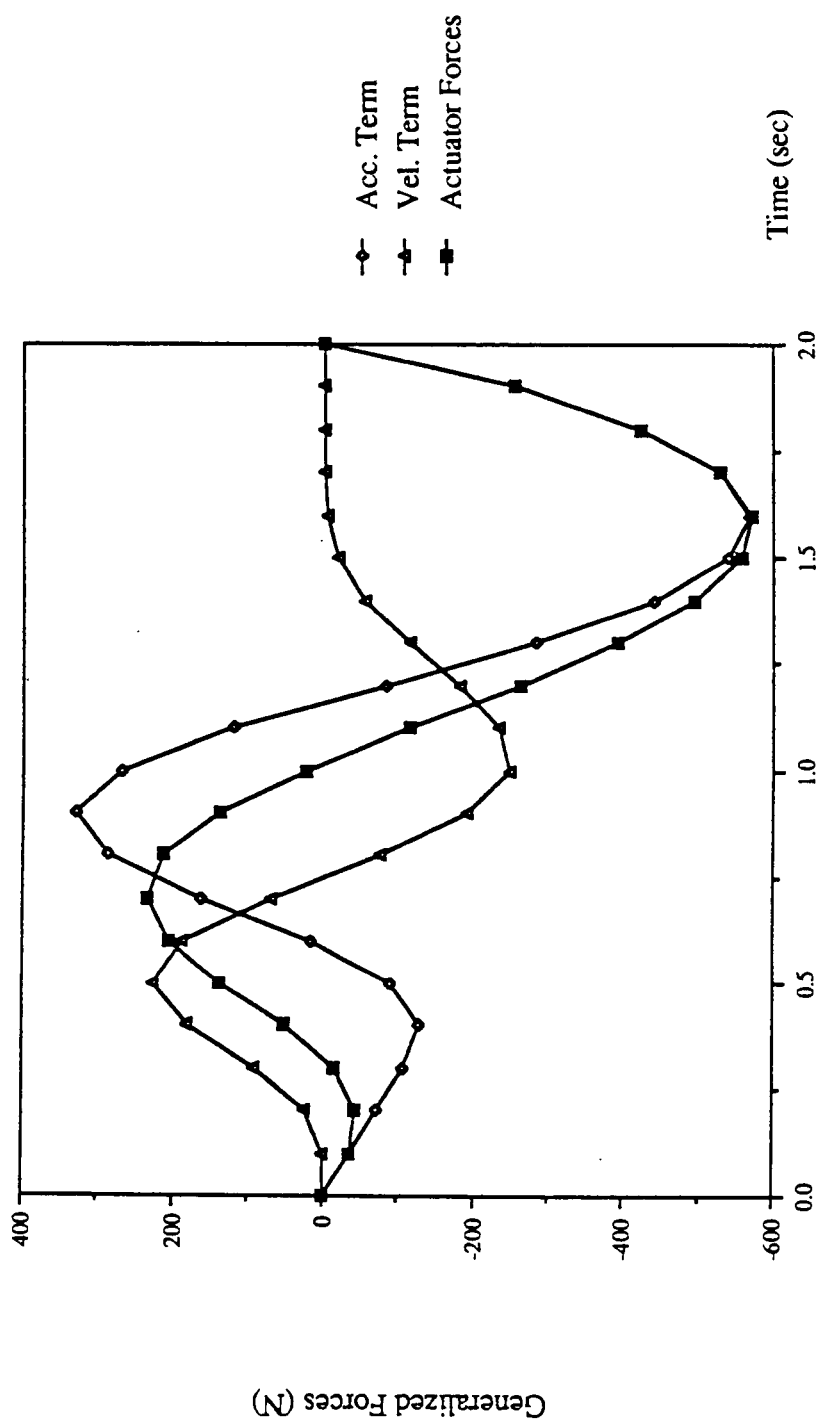


Fig. 6-25 Actuator Forces E5002b

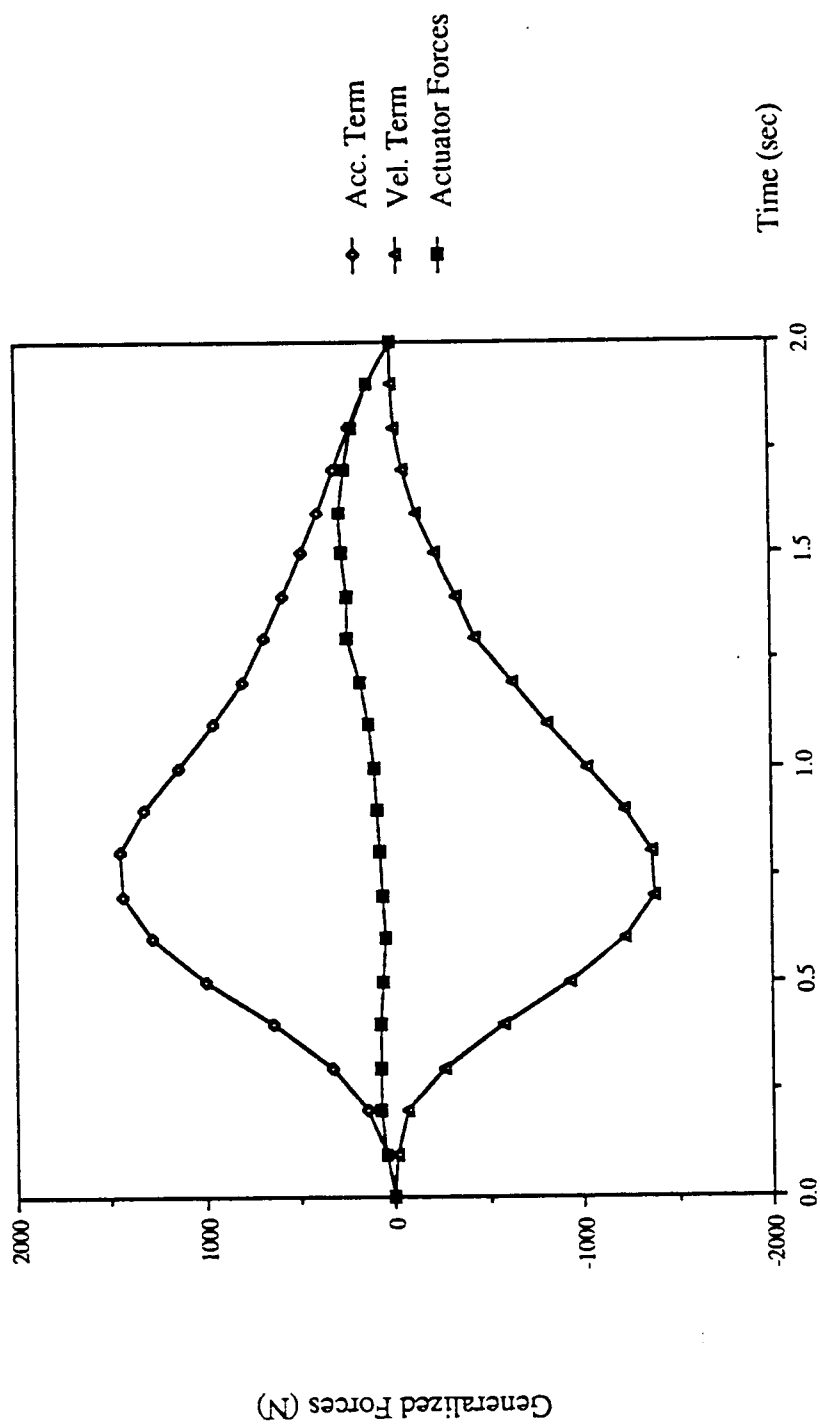


Fig. 6-26 Actuator Forces E5002c

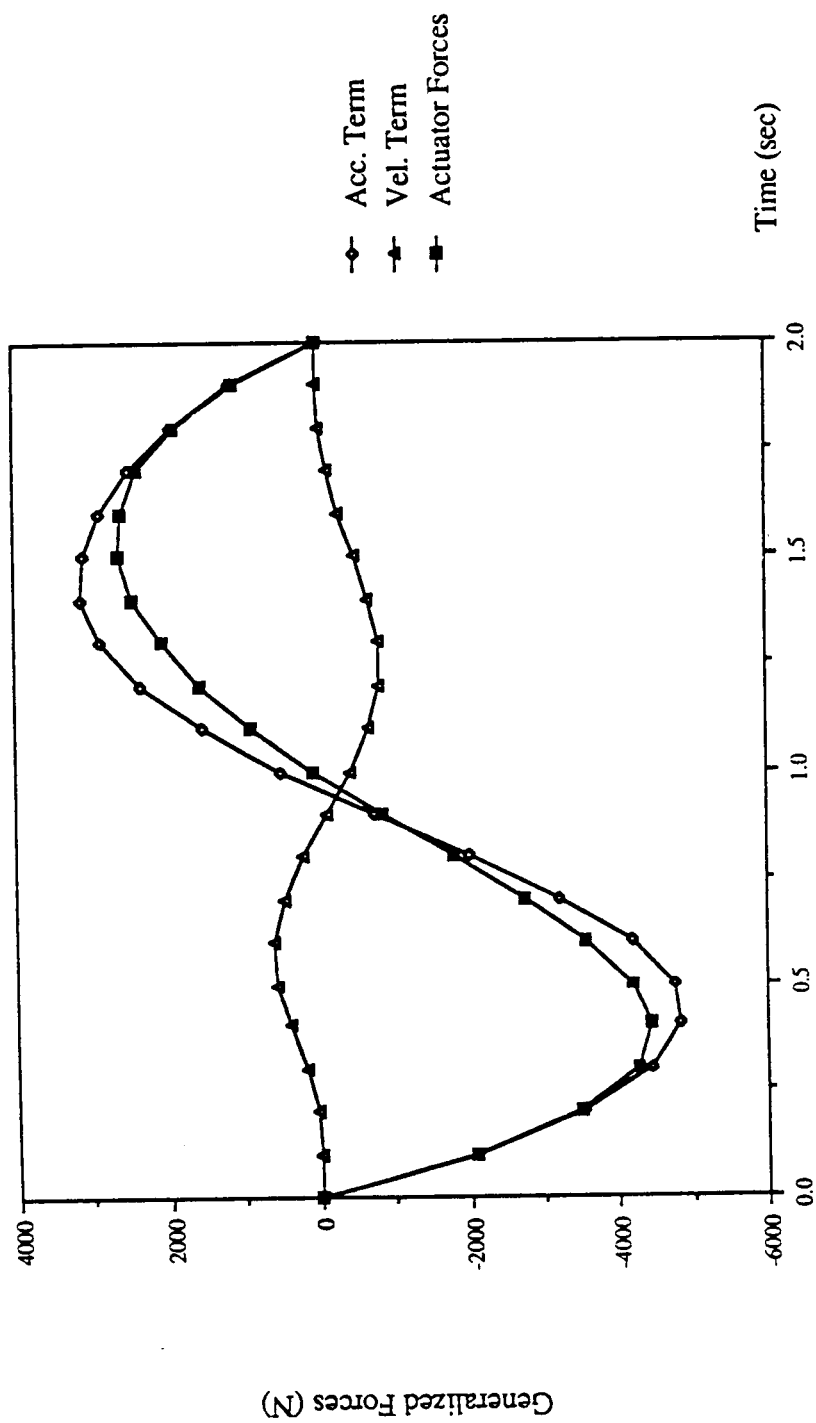


Fig. 6-27 Actuator Forces E5002d

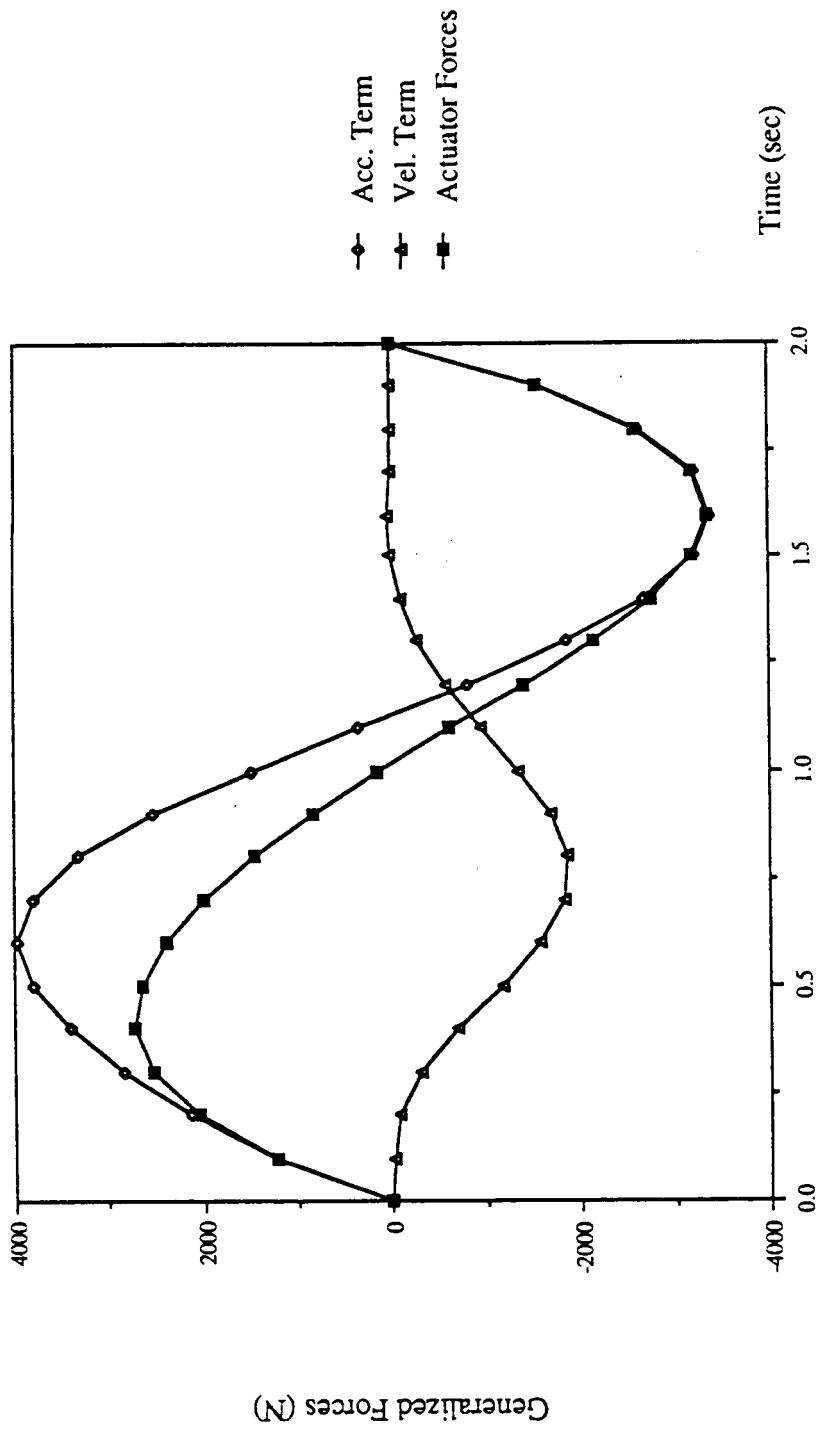


Fig. 6-28 Actuator Forces E5002c

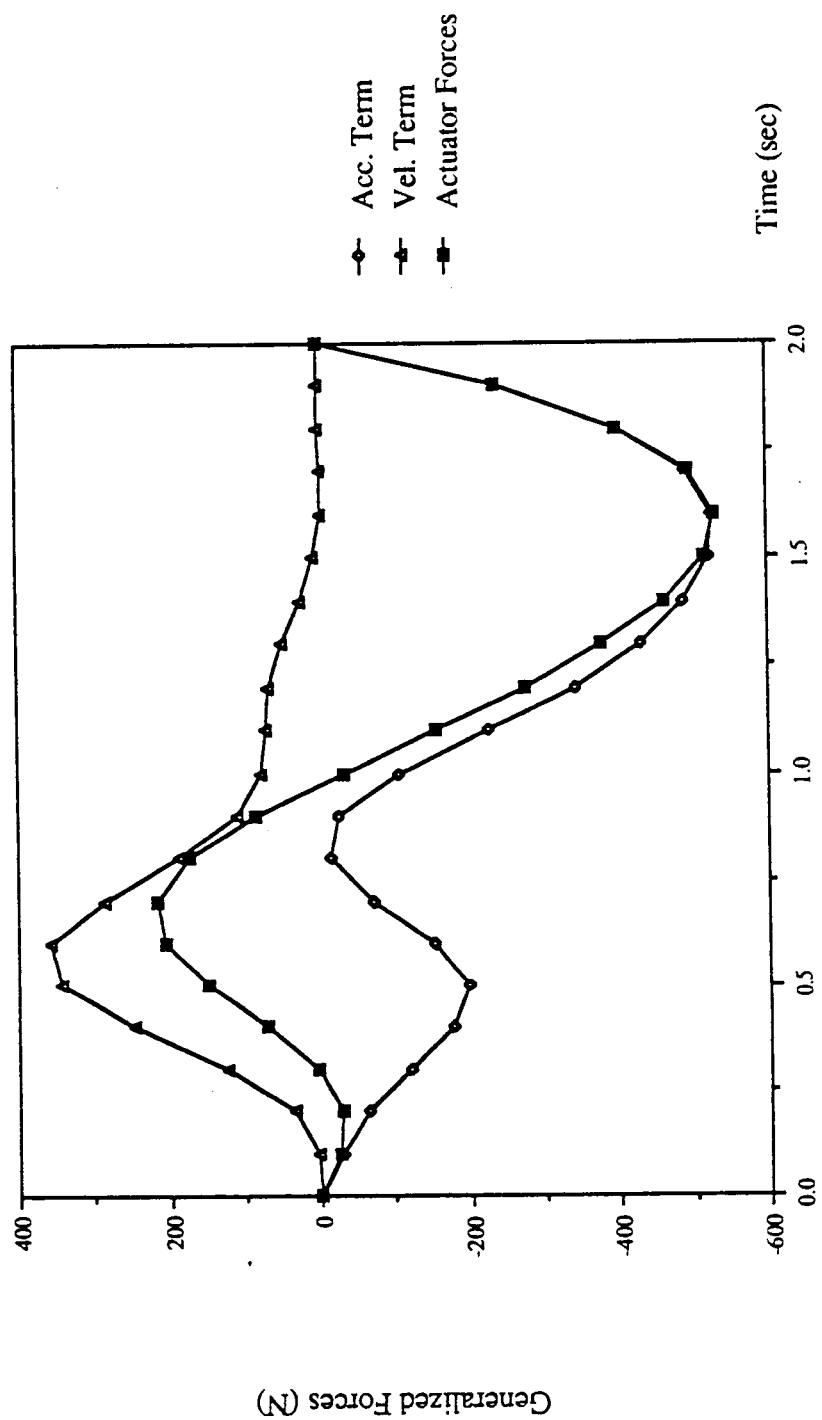


Fig. 6-29 Actuator Forces E5002f

$$\dot{\underline{d}} = \left[ G_u^d \right] \dot{\underline{U}} \quad (6-24)$$

$$\ddot{\underline{d}} = \left[ G_u^d \right] \ddot{\underline{U}} + \dot{\underline{U}}^T \left[ H_{uu}^d \right] \dot{\underline{U}} \quad (6-25)$$

the G- and H-functions are nonlinear, as expected. This is particularly obvious by comparing Fig. 6-1 for the platform motion with Fig. 6-8 and 6-10 for the actuator motion. When the platform is moving with constant acceleration and deceleration, the actuators are moving with nonlinear acceleration and deceleration. Notice that for "D" trajectory, only leg 1 and leg 2 (ie., "a" and "b") are plotted. This is because this trajectory is symmetric about the X\* axis. Hence, all the odd number legs are the same, as are all the even number legs.

Next, looking at the plots for generalized forces versus time for each of the motions, as expected, at the beginning of the motions when the velocities are small, the acceleration related forces dominate the required generalized forces for the actuators. Overall, the acceleration related forces are generally more significant than the velocity related forces. This is because the elements of the effective inertia matrix  $\left[ \dot{I}_{dd} \right]$  generally have larger magnitude than that of the elements for the inertia power array  $\left[ \dot{P}_{ddd} \right]$ . Furthermore, it is observed from Fig. 6-7, 9, 11, ..., 21 that  $\dot{\underline{d}}$  remains relatively small, less than 1m/s, with respect to  $\ddot{\underline{d}}$ . Recalling equation 5-59,

$$\underline{T}_d = \left[ \dot{I}_{dd} \right] \ddot{\underline{d}} + \dot{\underline{d}}^T \left[ \dot{P}_{ddd} \right] \dot{\underline{d}} \quad (5-59)$$

for the generalized forces, since  $\left[ \dot{P}_{ddd} \right]$  is operated on quadratically by  $\dot{\underline{d}}$ , the velocity related term is expected to be smaller than the acceleration related term. In addition, examining the configuration dependent effective inertia matrix  $\left[ \dot{I}_{dd} \right]$ , reveals that it is symmetric, which is consistent with its definition, and the magnitude of the diagonal elements are generally larger than the off-diagonal elements. This implies that  $\left[ \dot{I}_{dd} \right]$  has the tendency to decouple, although not completely. An example of  $\left[ \dot{I}_{dd} \right]$  is shown in Table 6-2. Table 6-2 also shows an

$$[\dot{I}_{dd}] = \begin{bmatrix} 1288.6 & 334.8 & -114.2 & 472.9 & -195.4 & -856.4 \\ 334.8 & 1622.9 & -919.5 & -516.3 & 536.0 & -535.3 \\ -114.2 & -919.5 & 1145.7 & 370.3 & -52.4 & 500.5 \\ 472.9 & -516.3 & 370.3 & 1631.2 & -891.9 & -543.6 \\ -195.4 & 536.0 & -52.4 & -891.9 & 1226.9 & 307.2 \\ -856.4 & -535.3 & 500.5 & -543.6 & 307.2 & 1650.1 \end{bmatrix}$$

$$[\dot{P}_{ddd}]_{i,j} = \begin{bmatrix} -172.0 & 75.1 & 20.9 & 9.7 & -73.7 & 131.3 \\ 25.0 & -423.6 & 85.4 & -37.4 & 53.3 & -8.1 \\ 220.2 & 58.7 & 15.6 & 1.6 & -152.2 & 100.0 \\ -122.5 & -11.7 & 77.5 & -120.4 & 239.3 & -82.4 \\ -62.6 & -5.9 & 32.9 & 108.8 & -96.6 & -53.2 \\ -2.0 & 18.3 & 18.6 & 24.8 & 53.8 & -83.2 \end{bmatrix}$$

**Table 6-2 Examples of Effective Inertia Matrix and a Plane of Inertia Power Array for the Generalized Stewart Platform**

example of one of the plane of the inertia power array  $\begin{bmatrix} \dot{P}_{ddd} \end{bmatrix}$ . Reviewing the general form of the inertia power array indicates that it is not decoupled nor does it have the tendency to decouple. The characteristics of the  $\begin{bmatrix} \dot{I}_{dd} \end{bmatrix}$  matrix and  $\begin{bmatrix} \dot{P}_{ddd} \end{bmatrix}$  array give some very important information concerning the controlling equation for the implementation of real-time feedforward control, which will be discussed shortly.

This work provides all the essential information for two different control strategies (Fu, Gonzalez and Lee [17], Luh, Walker and Paul [27], Ogata [32], Whitney [48]) for the generalized Stewart Platform( or DDTS ). To employ pure, non-dynamic, feedback control for the DDTS, merely specify the desired platform motion, the program will calculate the kinematic state of the platform at each time step. Subsequently, the required actuator lengths corresponding to the respective time step are calculated. Hence, in each segment of time, the initial and final positions of the actuators are known. This gives the necessary information for any of a number of position control algorithms.

The main objective of this work is, however, to provide the necessary software for future study using the dynamic model to develop a control scheme, possibly the computed torque technique, for the DDTS. Basically the computed torque technique can be slightly altered to become a layered control having both feedforward and non-linear feedback components. The interaction forces among all the various joints are compensated by the control components. The feedback component is used to compute the necessary correction torque to compensate for any deviation from the desired trajectory. Due to the computational demands of this control scheme, for any real-time control, it maybe necessary to simplify the controlling equations. It is customary although not always acceptable, to neglect the velocity-related coupling terms and the off-diagonal terms of the acceleration-related matrix for closed-loop control of serial manipulators. But, very little has been mentioned for parallel manipulators. Hence, this work provides some insights for the control of parallel manipulators.

Investigating the  $\begin{bmatrix} \dot{I}_{dd} \end{bmatrix}$  matrix and the  $\begin{bmatrix} \dot{P}_{ddd} \end{bmatrix}$  array of the Stanford manipulator, generally shows that the inertia matrix is highly decoupled, the off-

diagonal elements are either zero or much smaller than the diagonal elements, and the elements in the  $\begin{bmatrix} \dot{P}_{ddd} \end{bmatrix}$  array are mainly zero. Examples of the  $\begin{bmatrix} \dot{I}_{dd} \end{bmatrix}$  matrix and the  $\begin{bmatrix} \dot{P}_{ddd} \end{bmatrix}$  array for the Stanford manipulator are shown in Table 6-3. This observation tends to support the simplification scheme adopted for some serial manipulators as mentioned before. However, this scheme may not work effectively for parallel manipulators since the  $\begin{bmatrix} \dot{I}_{dd} \end{bmatrix}$  matrix for parallel manipulators is not as decoupled as it is for some serial manipulators and  $\begin{bmatrix} \dot{P}_{ddd} \end{bmatrix}$  array is relatively significant in the parallel case. As a result, employing the same simplification scheme to the parallel manipulators is likely to result in larger error, subsequently relying more heavily on feedback control.

The investigation conducted to study the effect of the platform mass relative to the leg mass in the effective inertia matrix and the inertia power array shows that with only the platform having mass, the resulting elements in the  $\begin{bmatrix} \dot{I}_{dd} \end{bmatrix}$  matrix generally have the same order of magnitude, although the main diagonal elements are slightly larger than the others. On the other hand, when the six legs have mass while the platform is massless, the main diagonal elements in the  $\begin{bmatrix} \dot{I}_{dd} \end{bmatrix}$  matrix are generally larger than the off-diagonal elements by an order of magnitude. These two cases indicate that the platform has a larger coupling effect than the legs in the final dynamic model. This provides information for one of the essential factors in the design of parallel manipulators. The information indicates that the  $\begin{bmatrix} \dot{I}_{dd} \end{bmatrix}$  matrix has the tendency to decouple when the mass of the legs increases, for the purpose of increasing the rigidity, relative to the platform. Unfortunately, the inertia power array does not show any noticeable general tendency. Examples of the  $\begin{bmatrix} \dot{I}_{dd} \end{bmatrix}$  matrix and  $\begin{bmatrix} \dot{P}_{ddd} \end{bmatrix}$  array are shown in Table 6-4 and Table 6-5 for the case when the legs are massless and when the platform is massless, respectively.

It is not the intention of this work to recommend the control strategy to use for parallel manipulators. However, this work provides the essential model and software for further study of parallel manipulators. To arrive at an efficient and relatively accurate control scheme for real-time control of the generalized Stewart

$$[I_{dd}] = \begin{bmatrix} 26920.0 & 0.0 & -570.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 23045.0 & 0.0 & 0.0 & 10.0 & 0.0 \\ -570.0 & 0.0 & 95.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 15.0 & 0.0 & 0.0 \\ 0.0 & 10.0 & 0.0 & 0.0 & 10.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

$$[P_{ddd}]_{i,j} = \begin{bmatrix} 0.0 & -5.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ -15.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ -10.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \\ 0.0 & 0.0 & 0.0 & 0.0 & 0.0 & 0.0 \end{bmatrix}$$

**Table 6-3 Examples of Effective Inertia Matrix and a Plane of Inertia Power Array for the Stanford Manipulator**

$$[I_{dd}^*] = \begin{bmatrix} 781.9 & 398.5 & -47.2 & 412.4 & -364.2 & -514.4 \\ 398.5 & 866.5 & -573.8 & -249.4 & 450.2 & -340.0 \\ -47.2 & -573.8 & 784.6 & 386.4 & -346.0 & 484.2 \\ 412.4 & -249.4 & 386.4 & 836.8 & -476.9 & -232.5 \\ -364.2 & 450.2 & -346.0 & -476.9 & 923.1 & 330.3 \\ -514.4 & -340.0 & 484.2 & -232.5 & 330.3 & 819.6 \end{bmatrix}$$

$$[P_{dd}^*]_{i,j} = \begin{bmatrix} 35.5 & -75.4 & -89.8 & 71.1 & 134.8 & 56.6 \\ 2.4 & 20.3 & 158.6 & -45.8 & -76.1 & -46.1 \\ -21.0 & 85.2 & -69.7 & -50.6 & 31.6 & -67.4 \\ 28.0 & 50.0 & -6.5 & -5.9 & -29.0 & 51.4 \\ 60.0 & -45.4 & -11.9 & 46.2 & -176.1 & 37.1 \\ 28.2 & -16.8 & -68.3 & 70.2 & 20.1 & -220.9 \end{bmatrix}$$

**Table 6-4 Examples of Effective Inertia Matrix and a Plane of Inertia Power Array for the Case when the Platform has Mass but the Legs are Massless**

$$[I_{dd}^*] = \begin{bmatrix} 388.7 & 31.8 & -5.5 & 55.2 & -58.4 & -165.3 \\ 31.8 & 460.2 & -169.9 & -39.8 & 45.3 & -86.8 \\ -5.5 & -169.9 & 397.2 & 61.5 & -70.7 & 42.3 \\ 55.2 & -39.8 & 61.5 & 397.4 & -154.6 & -41.4 \\ -58.4 & 45.3 & -70.7 & -154.6 & 414.7 & 46.8 \\ -165.3 & -86.8 & 42.3 & -41.4 & 46.8 & 444.5 \end{bmatrix}$$

$$[P_{ddd}^*] = \begin{bmatrix} 27.6 & -22.2 & -19.5 & -13.0 & 11.8 & 9.1 \\ -44.9 & 62.2 & 11.7 & 19.4 & 17.2 & -15.0 \\ -43.2 & 13.0 & -72.8 & 25.6 & 25.8 & -31.5 \\ 3.8 & -3.4 & -10.2 & -27.7 & -9.9 & 20.6 \\ 39.8 & -0.5 & 13.8 & -11.1 & -13.0 & 1.9 \\ 33.4 & -1.5 & -6.1 & 2.3 & -22.1 & 3.5 \end{bmatrix}$$

**Table 6-5 Examples of Effective Inertia Matrix and a Plane of Inertia Power Array for the Case when the Legs have Mass But the Platform is Massless**

Platform, additional research focusing on different trajectories and motion specifications needs to be done.

#### 6-4      Summary

In the second chapter, a general overview of the parallel manipulators classified as generalized Stewart Platforms was presented. The basic design concept and the control of the mechanism originated by Mr. D. Stewart was discussed briefly. Subsequently, various applications that stemmed from the original concept were discussed along with the different approaches taken by various researchers to analyze the mechanism.

The modeling technique adopted in this work, called the method of Kinematic Influence Coefficients, was addressed in Chapter 3. The notational scheme employed to facilitate the derivation of the model was also introduced. The presentation focused on modeling open-loop kinematic chains (or serial manipulators). The discussion included forward and reverse kinematic and the development of the dynamic model of serial manipulators.

Chapter 4 concentrated on the development of an isomorphic transformation technique called the transfer of generalized coordinates. This chapter started with the discussion of kinematic and dynamic model transfer for serial manipulators and then further extended the technique to include multi-loop parallel mechanisms. A general procedure was developed to perform the transfer of system dependence from any initial set of coordinates to the desired set of generalized coordinates.

Utilizing the modeling technique developed in the previous chapters, Chapter 5 established a complete model of the Dynamic Docking Test System. Starting from initializing the Denavit-Hartenberg parameters to the final desired dynamic model referenced to the common platform coordinates set, through a number of intermediate sets of generalized coordinates, this chapter described the procedures used to arrive at the final model for the computer simulation.

As a note of advice for similar simulation programs in the future, it is felt that FORTRAN language is inefficient and cumbersome for the modeling

technique adopted in this work due to the many matrices and higher dimensional arrays involved in the operation. A more efficient computer language suitable for matrix operation (eg., APL) will reduce the time and effort spent in coding and debugging the program. However, APL may not be computationally efficient for real-time control.

## APPENDIX A

### DEVELOPMENT AND DEFINITION OF GENERALIZED SCALAR ( • ) PRODUCT OPERATOR FUNDAMENTAL TO DYNAMIC MODELING AND TRANSFER OF COORDINATES

1. Quadratic operation of a *MATRIX* on a three dimensional array.

Given:

$$[A] = M \times N \quad \text{Matrix}$$

$$[B] = N \times M \times M \quad \text{Array}$$

Define:

$$[A]^T [B] [A] \equiv \begin{bmatrix} [A]^T [B]_{1,,:} [A] \\ [A]^T [B]_{2,,:} [A] \\ [A]^T [B]_{N,,:} [A] \end{bmatrix} = N \times N \times N \text{ Array}$$

2. Quadratic operation of a *VECTOR* on a three dimensional array.

Given:

$$\underline{d} = M \text{ component column vector}$$

Define:

$$\underline{d}^T [B] \underline{d} \equiv \begin{pmatrix} \underline{d}^T [B]_{1,,:} \underline{d} \\ \underline{d}^T [B]_{2,,:} \underline{d} \\ \underline{d}^T [B]_{N,,:} \underline{d} \end{pmatrix} = N \times 1 \text{ vector}$$

### 3. Vector multiplication of quadratic result

Given:

$$[C] = K \times N \quad \text{matrix}$$

$$[C]_k = k^{\text{th}} \text{ row of } [C]$$

Then

$$\begin{aligned} [C]_k (\underline{d}^T [B] \underline{d}) &= [C]_{k,1} \underline{d}^T [B]_{1,;} \underline{d} + [C]_{k,2} \underline{d}^T [B]_{2,;} \underline{d} \\ &\quad + \dots + [C]_{k,N} \underline{d}^T [B]_{N,;} \underline{d} \\ &= \underline{d}^T [C]_{k,1} [B]_{1,;} \underline{d} + \underline{d}^T [C]_{k,2} [B]_{2,;} \underline{d} \\ &\quad + \dots + \underline{d}^T [C]_{k,N} [B]_{N,;} \underline{d} \\ &= \underline{d}^T \left\{ \sum_{n=1}^N [C]_{k,n} [B]_{n,;} \right\} \underline{d} \\ &= \text{scalar} \end{aligned}$$

Define operator  $(\bullet)$  "DOT"

$$\begin{aligned} [C]_k \bullet [B] &\equiv \sum_{n=1}^N [C]_{k,n} [B]_{n,;} = M \times M \text{ matrix} \\ &= \text{scalar multiplication of planes followed by a} \\ &\quad \text{summation of the resulting planes} \end{aligned}$$

### 4. Matrix multiplication of quadratic result using $(\bullet)$ operator

$$[C] (\underline{d}^T [B] \underline{d}) = \begin{pmatrix} \underline{d}^T ([C]_{1,} \bullet [B]) \underline{d} \\ \underline{d}^T ([C]_{2,} \bullet [B]) \underline{d} \\ \vdots \\ \underline{d}^T ([C]_{k,} \bullet [B]) \underline{d} \end{pmatrix}$$

$$= \mathbf{d}^T ([\mathbf{C}] \cdot [\mathbf{B}]) \mathbf{d}$$

$$= K \times 1 \text{ vector}$$

where  $([\mathbf{C}] \cdot [\mathbf{B}]) = K \times M \times M$  array

**APPENDIX B**  
**SIMULATION PROGRAM LISTING**

```

PROGRAM STEWART_PLATFORM
C  ALL UNITS ARE IN SI
C  Y AND Z POSITIONS FIXED
C  X MOVED FROM Xi(m) TO Xf(m) AND ORIENTATION FROM Ai
C    TO Af(DEGREES)
C  THE NOTATIONS USED IN THE PROGRAM MOSTLY
C    CORRESPOND TO THE DERIVATION IN THE THESIS,
C    EXCEPT THE FINAL DYNAMIC MODEL WHERE INSTEAD OF
C    THE SUBSCRIPT "d" AS IN THE THESIS, THIS PROGRAM
C    USED "q"
REAL L3,L,II(3,3),Izz1,Izz2,Ixx2,Ixx3,Iyy3,M23,M34,Lc,Lr
DIMENSION P_PL(6,3),P_GL(6,3),P_LK(6,3),
+          ZSIphi_LK(3,3),ZPphi_LK(3,3,3),PF_Iuu(6,6),
+          PF_Puuu(6,6,6),ZSIphi(3,3),ZPphi(3,3,3),
+          ZGp_LK(3,3),ZHpp_LK(3,3,3),ZGp_phi(3,3),
+          ZHp_phi(3,3,3),PHI3(100,6),Q(6,1),UDOT(6,1),
+          UDDOT(6,1),QDOT(6,1),QDDOT(6,1)
COMMON/ORIEN/X1,Y1,Z1,X12,Y12,Z12
COMMON/ARR_Gu/Gu(6,3,6)
COMMON/ARR_Huu/Huu(6,3,6,6)
COMMON/DIRECT/DGp(6,3,3),DHpp(6,3,3,3)
COMMON/G_Hp_phi/Gp_phi(6,3,3),Hp_phi(6,3,3,3)
COMMON/ARR_phi_uu/Gphi_u(6,3,6),Hphi_uu(6,3,6,6)
COMMON/SINERTIA_phi/SIphi_LK(6,3,3)
COMMON/POWER_phi/Pphi_LK(6,3,3,3)
COMMON/SINERTIA_u/SIuu(6,6,6)
COMMON/POWER_u/Puuu(6,6,6,6)
COMMON/ARR_Gq_u/Gq_u(6,6)
COMMON/ARR_Hq_uu/Hq_uu(6,6,6)
COMMON/Iuu/TOT_Iuu(6,6)
COMMON/Puuu/TOT_Puuu(6,6,6)
COMMON/Iqq/STAR_Iqq(6,6)

```

```

COMMON/Pqqq/P_STAR_qqq(6,6,6)
COMMON/CGpHpp/L3
C   READ IN ALL THE NECESSARY DATA
CALL READ_DATA(X1,Y1,Z1,X12,Y12,Z12,XU,YU,ZU,
+           X1f,Y1f,Z1f,X12f,Y12f,Z12f,XUf,YUf,ZUf,
+           R,B,Izz1,Izz2,Ixx2,Ixx3,Iyy3,M23,M34,Lc,Lr,
+           TMAX,NSTEP,THETA,THETAf)
C   ATTENTION: ANGLE THETAf IS SPECIFIED AS POSITIVE,
C               BUT THE ROTATION CAN ONLY BE NEGATIVE
C               BECAUSE OF THE ASSUMPTION THAT THE
C               INITIAL ANGLE ALWAYS STARTS FROM
C               0 DEGREE, WHICH IS WHEN THE a12 AXIS IS
C               PARALLELED TO THE BASE Y* AXIS.
THETAf=-THETAf*3.1415926/180.
XUi=XU
THETAi=THETA*3.1415926/180.
C   CALCULATE TIME STEP
TSTEP=TMAX/NSTEP
WRITE(1,*)'TIME STEP=',TSTEP
DO III=1,NSTEP+1
    PTIME=(III-1)*TSTEP
C   CALCULATE THE NEXT PLATFORM POSITION AND
C   ORIENTATION
CALL NEXT_MOTION(TMAX,PTIME,XUi,XUF,THETAi,
+           THETAf,XU,THETA, V,W,ACC,AL,
+           Y12,Z12)
WRITE(1,*)' '
WRITE(1,*)'PRESENT TIME=',PTIME
WRITE(1,55)XU,YU,ZU
WRITE(1,*)'VELOCITY OF PLATFORM=',V
WRITE(1,*)'ACCELERATION OF PLATFORM=',ACC
WRITE(1,66)X1,Y1,Z1

```

```

WRITE(1,77)X12,Y12,Z12
WRITE(1,*)'THETA=',THETA
C   CALCULATE THE POSITION OF POINT P IN PLATFORM
C   FRAME
CALL P_PLATFORM(R,P_PL)
C   GET THE ROTATION MATRIX TO ORIENTATE THE
C   PLATFORM TO THE BASE FRAME
CALL TRANSF_P_G
C   TRANSFORM ALL THE POINTS ON THE PLATFORM TO THE
C   BASE FRAME
DO I=1,6
  CALL P_GLOBAL(I,Xu,Yu,Zu,P_PL,P_GL)
ENDDO
DO I=1,6
C   TRANSFORM ALL THE POINTS ON THE PLATFORM TO
C   THEIR RESPECTIVE LEG FRAME
  CALL P_LINK(I,B,P_GL,P_LK)
C   CALCULATE THE G- AND H-FUNCTIONS FOR THE
C   PLATFORM
  CALL CAL_Gu(I)
  CALL CAL_Huu(I)
C   CALCULATE THE G- AND H-FUNCTIONS FOR THE LEGS
  CALL CAL_G_Hp_phi(I,P_LK(I,1),P_LK(I,2),P_LK(I,3),
+      ZGp_phi,ZHp_phi)
C   DIRECT TRANSFER OF Hp_phi
  CALL CAL_Hphi_pp(I,ZGp_phi,ZHp_phi)
ENDDO
C   TO DETERMINE Gphi_u AND Hphi_uu
CALL CAL_G_Hphi_uu
C   SET UP THE INERTIA MATRICES AND POWER ARRAY
DO I=1,6
  CALL CAL_INERTIA(I,Izz1,Izz2,Ixx2,Ixx3,Iyy3,

```

```

+           M23,M34,Lc,Lr,Siphi_LK)
      CALL CAL_POW_phi(I,Ixx2,Ixx3,Iyy3,M23,M34,
+           Lc,Lr,Pphi_LK)
      ENDDO
C      TO DETERMINE Iuu AND Puuu
      CALL CAL_Iuu_Puuu
C      CALCULATE THE PLATFORM INERTIA MATRIX AND POWER
C      ARRAY
      CALL PFORM_INERTIA(R,II,PF_Iuu)
      CALL PFORM_INEPOW(II,PF_Puuu)
C      CALCULATE THE INERTIA MATRIX AND POWER ARRAY
C      FOR THE SYSTEM
      CALL I_STAR_uu(PF_Iuu,TOT_Iuu)
      CALL P_STAR_uuu(PF_Puuu,TOT_Puuu)
C      DETERMINE THE FINAL DESIRED FIRST- AND SECOND-
C      ORDER KIC,Gq_u AND Hq_uu
      CALL CAL_Gq_u
      CALL CAL_Hq_uu
C      DETERMINE THE DESIRED INERTIA MATRIX AND POWER
C      ARRAY I*qq AND P*qqq
      CALL CAL_Iqq_Pqqq
C      SPECIFIED THE PLATFORM MOTION
      UDOT(3,1)=V
      UDOT(4,1)=W
      UDDOT(3,1)=ACC
      UDDOT(4,1)=AL
C      CALCULATE THE ACTUATORS MOTION
      CALL CAL_QDOT(UDOT,QDOT)
      CALL CAL_QDDOT(UDOT,UDDOT,QDDOT)
      WRITE(1,88)
      DO K=1,6
        WRITE(1,99) UDOT(K,1),UDDOT(K,1),QDOT(K,1),

```

```

+          QDDOT(K,1)
      ENDDO
C      CALCULATE THE REQUIRED GENERALIZED FORCES FROM
C      EACH ACTUATOR
      CALL GEN_FORCES(QDOT,QDDOT)
      ENDDO
55  FORMAT(X,'Xu=',F15.10,5X,'Yu=',F15.10,5X,'Zu=',F15.10)
66  FORMAT(X,'X1=',F15.10,5X,'Y1=',F15.10,5X,'Z1=',F15.10)
77  FORMAT(X,'X12=',F15.10,4X,'Y12=',F15.10,4X,'Z12=',
+      F15.10)
88  FORMAT(/,6X,'UDOT',14X,'UDDOT',13X,'QDOT',14X,'QDDOT')
99  FORMAT(X,4(F13.10,5X))
      STOP
      END

```

```

*****

```

```

*
*          SUBROUTINE READ_DATA
*
*

```

```

*****

```

```

      SUBROUTINE READ_DATA(X1i,Y1i,Z1i,X12i,Y12i,Z12i,XUi,
+      YUi,ZUi,X1f,Y1f,Z1f,X12f,Y12f,
+      Z12f,XUf,YUf,ZUf,R,B,Izz1,Izz2,
+      Ixx2, Ixx3,Iyy3,M23,M34,Lc,Lr,
+      TMAX,NSTEP,THETAi,THETAf)
      REAL Izz1,Izz2,Ixx2,Ixx3,Iyy3,M23,M34,Lc,Lr
C      READ IN INITIAL ORIENTATION OF THE PLATFORM
      READ (40,*) X1i,Y1i,Z1i,X12i,Y12i,Z12i,THETAi
C      READ IN INITIAL POSITION OF THE PLATFORM
      READ (40,*) XUi,YUi,ZUi
C      READ IN FINAL ORIENTATION OF THE PLATFORM
      READ (40,*) X1f,Y1f,Z1f,X12f,Y12f,Z12f,THETAf
C      READ IN FINAL POSITION OF THE PLATFORM

```

```

      READ (40,*) XUf,YUf,ZUf
C     READ IN PLATFORM AND BASE RADIUS
      READ (40,*) R,B
C     CALCULATE MASS MOMENT OF INERTIA
      CALL GET_CONST(Izz1,Izz2,Ixx2,Ixx3,Iyy3,M23,M34,Lc,Lr)
C     READ IN TOTAL TIME TAKEN
      READ (40,*) TMAX
C     READ IN NUMBER OF STEPS NEEDED
      READ (40,*) NSTEP
      RETURN
      END

```

```

*****

```

```

*
*
*           SUBROUTINE NEXT_MOTION
*

```

```

*****

```

```

      SUBROUTINE NEXT_MOTION(TMAX,PTIME,XUi,XUf,
+                               THETAi,THETAf,XU,THETA,V,
+                               W,ACC,AL,Y12,Z12)
C     THIS SUBROUTINE CALCULATES THE NEXT PLATFORM
C     MOTION USING D5002
      X=XUf-XUi
      TTA=THETAf-THETAi
      T=PTIME/TMAX
      XU=XUi+X*(10.*(T**3)-15.*(T**4)+6.*(T**5))
      V=(X/TMAX)*(30.*(T**2)-60.*(T**3)+30.*(T**4))
      ACC=(X/TMAX**2)*(60.*T-180.*(T**2)+120.*(T**3))
      THETA=THETAi+TTA*(10.*(T**3)-15.*(T**4)+6.*(T**5))
      W=(TTA/TMAX)*(30.*(T**2)-60.*(T**3)+30.*(T**4))
      AL=(TTA/TMAX**2)*(60.*T-180.*(T**2)+120.*(T**3))
      Y12=COS(THETA)
      Z12=SIN(THETA)

```

THETA=THETA\*180./3.1415926

RETURN

END

\*\*\*\*\*

\*

\*

\*

SUBROUTINE P\_PLATFORM

\*

\*

\*

\*\*\*\*\*

SUBROUTINE P\_PLATFORM(R,P)

C DETERMINE ALL THE POINTS ON THE PLATFORM IN TERMS

C OF THE PLATFORM FRAME

DIMENSION P(6,3)

C ASSUME THAT THE ANGLE BETWEEN THE POINTS TOGETHER

C IS 5 DEGREES

BETA=5.

P(1,1)=R

P(2,1)=R\*COSD(BETA)

P(2,2)=R\*SIND(BETA)

P(3,1)=R\*COSD(120.)

P(3,2)=R\*SIND(120.)

P(4,1)=R\*COSD(120.+BETA)

P(4,2)=R\*SIND(120.+BETA)

P(5,1)=R\*COSD(240.)

P(5,2)=R\*SIND(240.)

P(6,1)=R\*COSD(240.+BETA)

P(6,2)=R\*SIND(240.+BETA)

RETURN

END

```

*****
*
*                               SUBROUTINE P_GLOBAL
*
*****

```

```

      SUBROUTINE P_GLOBAL(I,Xu,Yu,Zu,P_P,P_G)
C    THIS SUBROUTINE TRANSFORMS ALL THE POINTS ON THE
C    PLATFORM TO THE BASE FRAME
      COMMON/ARR_R/ARM(6,3)
      COMMON/TRAN_P_G/R(3,3)
      DIMENSION P_P(6,3),P_G(6,3),P(3,1),AR(3,1)
      P(1,1)=P_P(I,1)
      P(2,1)=P_P(I,2)
      P(3,1)=P_P(I,3)
      CALL MATRIX_MULT(3,3,R,3,1,P,AR)
C    XP,YP,ZP IN GLOBAL COOR
      ARM(I,1)=AR(1,1)
      ARM(I,2)=AR(2,1)
      ARM(I,3)=AR(3,1)
      P_G(I,1)=Xu+ARM(I,1)
      P_G(I,2)=Yu+ARM(I,2)
      P_G(I,3)=Zu+ARM(I,3)
      RETURN
      END

```

```

*****
*
*                               SUBROUTINE P_LINK
*
*****

```

```

      SUBROUTINE P_LINK(I,B,P_G,P_L)
      DIMENSION P_G(6,3),P_L(6,3),P_LK(3,1),P(3,1),R(3,3),
+          RI(3,3)

```

```

C   XP,YP,ZP IN LEG COORDINATE FRAME
C   ASSUME THAT THE ANGLE BETWEEN THE TWO POINTS
C   TOGETHER IS 5 DEGREES
    DELTA=5.
    IF(I .EQ. 1) AL=0.
    IF(I .EQ. 2) AL=DELTA
    IF(I .EQ. 3) AL=120.
    IF(I .EQ. 4) AL=120.+DELTA
    IF(I .EQ. 5) AL=240.
    IF(I .EQ. 6) AL=240.+DELTA
    CAL=COSD(AL)
    SAL=SIND(AL)
C   POINT 1 ON THE BASE IS CONNECTED TO POINT 2 ON THE
C   PLATFORM AND SO ON
    J=I+1
    IF(I .EQ. 6) J=1
    P(1,1)=P_G(J,1)
    P(2,1)=P_G(J,2)-B*CAL
    P(3,1)=P_G(J,3)-B*SAL
    CALL ROT_MATRIX(I,R)
    CALL MATRIX_INV(3,R,RI)
    CALL MATRIX_MULT(3,3,RI,3,1,P,P_LK)
    DO K=1,3
        P_L(I,K)=P_LK(K,1)
    ENDDO
    RETURN
END

```

```

*****
*
*                               SUBROUTINE CAL_Gu
*
*****

      SUBROUTINE CAL_Gu(N)
      COMMON/ARR_Gu/Gu(6,3,6)
      COMMON/ARR_R/ARM(6,3)
C     PUT Gu IN MATRIX FORM
C     SIMILAR REASON AS IN SUBROUTINE P_LINK FOR THE
C     FOLLOWING STEPS
      J=N+1
      IF(N .EQ. 6)J=1
      Gu(N,1,1)=1.
      Gu(N,1,2)=0.
      Gu(N,1,3)=0.
      Gu(N,1,4)=0.
      Gu(N,1,5)=ARM(J,3)
      Gu(N,1,6)=-ARM(J,2)
      Gu(N,2,1)=0.
      Gu(N,2,2)=1.
      Gu(N,2,3)=0.
      Gu(N,2,4)=-ARM(J,3)
      Gu(N,2,5)=0.
      Gu(N,2,6)=ARM(J,1)
      Gu(N,3,1)=0.
      Gu(N,3,2)=0.
      Gu(N,3,3)=1.
      Gu(N,3,4)=ARM(J,2)
      Gu(N,3,5)=-ARM(J,1)
      Gu(N,3,6)=0.
      RETURN

```

END

\*\*\*\*\*

\* \*

\* SUBROUTINE CAL\_Huu \*

\* \*

\*\*\*\*\*

SUBROUTINE CAL\_Huu(N)

COMMON/ARR\_Huu/Huu(6,3,6,6)

COMMON/ARR\_R/ARM(6,3)

DATA Huu/648\*0./

C PUT Huu IN MATRIX FORM

C SIMILAR REASON AS BEFORE

J=N+1

IF(N.EQ. 6)J=1

Huu(N,1,4,5)=ARM(J,2)

Huu(N,1,4,6)=ARM(J,3)

Huu(N,1,5,5)=-ARM(J,1)

Huu(N,1,6,6)=-ARM(J,1)

C

Huu(N,2,4,4)=-ARM(J,2)

Huu(N,2,4,5)=ARM(J,1)

Huu(N,2,5,6)=ARM(J,3)

Huu(N,2,6,6)=-ARM(J,2)

C

Huu(N,3,4,4)=-ARM(J,3)

Huu(N,3,4,6)=ARM(J,1)

Huu(N,3,5,5)=-ARM(J,3)

Huu(N,3,5,6)=ARM(J,2)

RETURN

END

```

*****
*
*                               SUBROUTINE CAL_G_Hp_phi
*
*****

      SUBROUTINE CAL_G_Hp_phi(N,XP,YP,ZP,ZGp_phi,ZHp_phi)
      REAL L3,L
      DIMENSION ZGp_LK(3,3),ZHpp_LK(3,3,3),ZGp_phi(3,3),
+           ZHp_phi(3,3,3)
      COMMON/CGpHpp/L3,PSI1,PSI2,S1,S2,C1,C2,T2,TETA1,
+           TETA2
      COMMON/DIRECT/DGp(6,3,3),DHpp(6,3,3,3)
      PI=ACOS(-1.)
      L3=SQRT(XP**2+YP**2+ZP**2)
      PSI2=ACOS(ZP/L3)
      TETA2=PI+PSI2
      L=L3*SIN(PSI2)
      PSI1=ASIN(YP/L)
      TETA1=PI+PSI1
      S1=SIN(PSI1)
      S2=SIN(PSI2)
      C1=COS(PSI1)
      C2=COS(PSI2)
      T2=TAN(PSI2)
C      TWO DIFFERENT WAYS TO DETERMINE Gp_phi AND Hp_phi
C      THIS PROGRAM USED THE RESULTS FROM INDIRECT, BUT
C      THE RESULTS FROM DIRECT SERVED AS A CHECK
C      CALL CAL_DR_Gp(N,XP,YP,ZP,DGp)
C      CALL CAL_DR_Hpp(N,XP,YP,ZP,DGp,DHpp)
      CALL INDIRECT(N,ZGp_phi,ZHp_phi)
      RETURN
      END

```

```

*****
*
*
*
*
*****

```

# SUBROUTINE INDIRECT

```

SUBROUTINE INDIRECT(N,ZGp_phi,ZHp_phi)
C THIS SUBROUTINE DETERMINES Gp_phi AND Hp_phi
C INDIRECTLY
  DIMENSION Gs(3,3),Hss(3,3,3),ZGp_phi(3,3),
+           ZHp_phi(3,3,3)
  COMMON/G_Hp_phi/Gp_phi(6,3,3),Hp_phi(6,3,3,3)
  CALL CAL_Gs(N,ST1,ST2,CT1,CT2,Gs,ZGp_phi)
  CALL CAL_Hss(N,ST1,ST2,CT1,CT2,Gs,Hss,ZHp_phi)
  CALL CHG_DIMLO(N,3,3,ZGp_phi,3,3,3,ZHp_phi,Gp_phi,
+               Hp_phi)
  RETURN
  END

```

```

*****
*
*
*
*
*****

```

# SUBROUTINE CAL\_Hphi\_pp

```

SUBROUTINE CAL_Hphi_pp(N,ZGp_phi,ZHp_phi)
C THIS SUBROUTINE CALCULATES THE INVERSE OF Gp_phi
  AND Hp_phi
  COMMON/G_Hphi_pp/Gphi_p(6,3,3),Hphi_pp(6,3,3,3)
  DIMENSION ZGp_phi(3,3),ZHp_phi(3,3,3),ZGphi_p(3,3),
+           TZGphi_p(3,3),XX(3,3,3),YY(3,3,3),ZZ(3,3,3),
+           ZHphi_pp(3,3,3)
  CALL MATRIX_INV(3,ZGp_phi,ZGphi_p)
  CALL TRANSPOSE(3,3,ZGphi_p,TZGphi_p)
  CALL GENERAL_DOT(3,3,ZGphi_p,3,ZHp_phi,XX)

```

```

DO I=1,3
  CALL NN_DOT_NNN(3,3,I,3,TZGphi_p,XX,YY)
  CALL NNN_DOT_NN(3,3,I,3,YY,ZGphi_p,ZZ)
  DO J=1,3
    DO K=1,3
      ZHphi_pp(I,J,K)=-ZZ(I,J,K)
    ENDDO
  ENDDO
ENDDO
CALL CHG_DIMLO(N,3,3,ZGphi_p,3,3,3,ZHphi_pp,Gphi_p,
+
  Hphi_pp)
RETURN
END

```

```

*****

```

```

*
*
*
*
*
*

```

# SUBROUTINE CAL\_DR\_Gp

```

*****

```

```

SUBROUTINE CAL_DR_Gp(N,XP,YP,ZP,Gp)
C THIS SUBROUTINE IS NOT ACTUALLY USED BUT ONLY
C SERVES AS A CHECK
REAL L3,L
DIMENSION Gp(6,3,3)
COMMON/CGpHpp/L3,PSI1,PSI2,S1,S2,C1,C2,T2,TETA1,
+
  TETA2
A1=(L3**2)*T2
D2DXP=XP/A1
D2DYP=YP/A1
D2DZP=(ZP*C2-L3)/((L3**2)*S2)
A2=C1*(L3*S2)**2
D1DXP=-XP*S1/A2
D1DYP=(L3*S2-YP*S1)/A2

```

```

D1DZP=0
D3DXP=XP/L3
D3DYP=YP/L3
D3DZP=ZP/L3
Gp(N,1,1)=D1DXP
Gp(N,2,1)=D2DXP
Gp(N,3,1)=D3DXP
Gp(N,1,2)=D1DYP
Gp(N,2,2)=D2DYP
Gp(N,3,2)=D3DYP
Gp(N,1,3)=D1DZP
Gp(N,2,3)=D2DZP
Gp(N,3,3)=D3DZP
RETURN
END

```

```

*****
*
*                               SUBROUTINE CAL_DR_Hpp
*
*****

```

```

      SUBROUTINE CAL_DR_Hpp(N,XP,YP,ZP,Gp,Hpp)
C      THIS SUBROUTINE IS ALSO NOT USED BUT SERVES AS A
C      CHECK
      REAL L3,L
      DIMENSION Gp(6,3,3),Hpp(6,3,3,3)
      COMMON/CGpHpp/L3,PSI1,PSI2,S1,S2,C1,C2,T2,TETA1,
+      TETA2
      A2=C1*(L3*S2)**2
      D23D2XP=(L3**2-XP**2)/L3**3
      D23DXPDYP=-XP*YP/L3**3
      D23DXPDZP=-XP*ZP/L3**3
      D23DYDPXP=D23DXPDYP

```

$$\begin{aligned}
& D23D2YP = ((L3^{**2}) - (YP^{**2})) / L3^{**3} \\
& D23DYPDZP = -YP * ZP / L3^{**3} \\
& D23DZPDXP = D23DXPDZP \\
& D23DZPDYP = D23DYPDZP \\
& D23D2ZP = ((L3^{**2}) - (ZP^{**2})) / L3^{**3} \\
& A4 = (L3^{**4}) * (T2^{**3}) \\
& D22D2XP = (((L3 * T2)^{**2}) - 2 * ((XP * T2)^{**2}) - \\
+ & \quad (XP / C2)^{**2}) / A4 \\
& D22DXPDYP = (-XP * YP * (((1 / C2)^{**2}) + 2 * (T2^{**2}))) / A4 \\
& A5 = (L3^{**4}) * (T2^{**2}) * S2 \\
& D22DXPDZP = (-2 * XP * ZP * S2 * T2 - XP * ZP / C2 \\
+ & \quad + L3 * XP / (C2^{**2})) / A5 \\
& D22D2YP = ((L3 * T2)^{**2} - 2 * ((YP * T2)^{**2}) - (YP / C2)^{**2}) / A4 \\
& D22DYPDXP = D22DXPDYP \\
& D22DYPDZP = (-2 * YP * ZP * S2 * T2 - YP * (ZP * C2 - \\
+ & \quad L3) / (C2^{**2})) / A5 \\
& D22DZPDXP = D22DXPDZP \\
& D22DZPDYP = D22DYPDZP \\
& D22D2ZP = ((S2 * L3^{**2}) * (-Gp(N, 2, 3) * ZP * S2 + C2 - ZP / L3) - \\
& \quad (ZP * C2 - L3) * (2 * ZP * S2 + Gp(N, 2, 3) * C2 * L3^{**2})) \\
+ & \quad / ((S2 * L3^{**2})^{**2}) \\
& A6 = 2 * C1 * S2^{**2} \\
& A7 = XP * C1 \\
& A8 = XP * S1 \\
& A9 = 2 * C1 * C2 * S2 * L3^{**2} \\
& A10 = S1 * (S2 * L3)^{**2} \\
& A11 = S1 * S2^{**2} \\
& A12 = L3 * C2 \\
& A13 = L3 * S2 - YP * S1 \\
& D21D2XP = -(A2 * (Gp(N, 1, 1) * A7 + S1) - A8 * (XP * A6 + \\
+ & \quad Gp(N, 2, 1) * A9 - Gp(N, 1, 1) * A10)) / A2^{**2} \\
& D21DXPDYP = -(A2 * Gp(N, 1, 2) * A7 - A8 * (YP * A6 +
\end{aligned}$$

$$\begin{aligned}
& + \frac{\text{Gp}(N,2,2)*A9 - \text{Gp}(N,1,2)*A10)}{A2^{**2}} \\
& \text{D21DXPDZP} = -(A2*\text{Gp}(N,1,3)*A7 - A8*(ZP*A6 + \\
& + \frac{\text{Gp}(N,2,3)*A9 - \text{Gp}(N,1,3)*A10)}{A2^{**2}} \\
& \text{D21D2YP} = (A2*(YP*S2/L3 + A12*\text{Gp}(N,2,2) - S1 - \\
& + \frac{\text{Gp}(N,1,2)*YP*C1) - A13*(YP*A6 + \text{Gp}(N,2,2)*A9 - \\
& + \frac{\text{Gp}(N,1,2)*A10)}{A2^{**2}} \\
& \text{D21DYPDZP} = (A2*(ZP*S2/L3 + A12*\text{Gp}(N,2,3) - \\
& + \frac{\text{Gp}(N,1,3)*YP*C1) - A13*(ZP*A6 + \\
& + \frac{\text{Gp}(N,2,3)*A9 - \text{Gp}(N,1,3)*A10)}{A2^{**2}}
\end{aligned}$$

$$\text{D21D2ZP} = 0$$

$$\text{D21DZPDXP} = 0$$

$$\text{D21DZPDYP} = 0$$

$$\text{D21DYPDXP} = \text{D21DXPDYP}$$

$$\text{Hpp}(N,1,1,1) = \text{D21D2XP}$$

$$\text{Hpp}(N,1,1,2) = \text{D21DXPDYP}$$

$$\text{Hpp}(N,1,1,3) = \text{D21DXPDZP}$$

$$\text{Hpp}(N,1,2,1) = \text{D21DYPDXP}$$

$$\text{Hpp}(N,1,2,2) = \text{D21D2YP}$$

$$\text{Hpp}(N,1,2,3) = \text{D21DYPDZP}$$

$$\text{Hpp}(N,1,3,1) = \text{D21DZPDXP}$$

$$\text{Hpp}(N,1,3,2) = \text{D21DZPDYP}$$

$$\text{Hpp}(N,1,3,3) = \text{D21D2ZP}$$

\*

$$\text{Hpp}(N,2,1,1) = \text{D22D2XP}$$

$$\text{Hpp}(N,2,1,2) = \text{D22DXPDYP}$$

$$\text{Hpp}(N,2,1,3) = \text{D22DXPDZP}$$

$$\text{Hpp}(N,2,2,1) = \text{D22DYPDXP}$$

$$\text{Hpp}(N,2,2,2) = \text{D22D2YP}$$

$$\text{Hpp}(N,2,2,3) = \text{D22DYPDZP}$$

$$\text{Hpp}(N,2,3,1) = \text{D22DZPDXP}$$

$$\text{Hpp}(N,2,3,2) = \text{D22DZPDYP}$$

$$\text{Hpp}(N,2,3,3) = \text{D22D2ZP}$$

```

*
  Hpp(N,3,1,1)=D23D2XP
  Hpp(N,3,1,2)=D23DXPDYP
  Hpp(N,3,1,3)=D23DXPDZP
  Hpp(N,3,2,1)=D23DYPDXP
  Hpp(N,3,2,2)=D23D2YP
  Hpp(N,3,2,3)=D23DYPDZP
  Hpp(N,3,3,1)=D23DZPDXP
  Hpp(N,3,3,2)=D23DZPDYP
  Hpp(N,3,3,3)=D23D2ZP
  RETURN
  END
*****
*
*
*
*
*****
      SUBROUTINE CAL_Gs
*****
      SUBROUTINE CAL_Gs(N,ST1,ST2,CT1,CT2,Gs,Gphi)
C      THIS SUBROUTINE CALCULATES THE G-FUNCTION FOR
C      EACH LEG
      REAL L3
      COMMON/CGpHpp/L3,PSI1,PSI2,S1,S2,C1,C2,T2,TETA1,
+      TETA2
      COMMON/TETA_L/ST(6),CT(6),RL3(6)
      DIMENSION Gs(3,3),R(3,3),Gphi(3,3)
      WRITE(1,*)'ACTUATOR LENGTH L3=',L3
      ST1=SIN(TETA1)
      ST2=SIN(TETA2)
      CT1=COS(TETA1)
      CT2=COS(TETA2)
      ST(N)=ST2
      CT(N)=CT2

```

```

      RL3(N)=L3
      Gs(1,1)=-L3*ST1*ST2
      Gs(1,2)=L3*CT1*CT2
      Gs(1,3)=CT1*ST2
      Gs(2,1)=L3*CT1*ST2
      Gs(2,2)=L3*ST1*CT2
      Gs(2,3)=ST1*ST2
      Gs(3,1)=0
      Gs(3,2)=L3*ST2
      Gs(3,3)=-CT2
C     TRANSFORM THE G-FUNCTION FROM THE LEG FRAME TO THE
C     BASE FRAME
      CALL ROT_MATRIX(N,R)
      CALL MATRIX_MULT(3,3,R,3,3,Gs,Gphi)
      RETURN
      END

*****
*                                     *
*                                     *
*          SUBROUTINE CAL_Hss          *
*                                     *
*****

      SUBROUTINE CAL_Hss(N,ST1,ST2,CT1,CT2,Gs,Hss,Hphi)
C     THIS SUBROUTINE CALCULATES THE H-FUNCTION FOR
C     EACH LEG
      REAL L3
      DIMENSION Hss(3,3,3),Gs(3,3),R(3,3),Hphi(3,3,3)
      COMMON/CGpHpp/L3,PSI1,PSI2,S1,S2,C1,C2,T2,TETA1,
+          TETA2
      Hss(1,1,1)=-Gs(2,1)
      Hss(1,1,2)=-Gs(2,2)
      Hss(1,1,3)=-Gs(2,3)
      Hss(1,2,1)=Hss(1,1,2)

```

```

Hss(1,2,2)=Hss(1,1,1)
Hss(1,2,3)=CT1*CT2
Hss(1,3,1)=Hss(1,1,3)
Hss(1,3,2)=Hss(1,2,3)
Hss(2,1,1)=Gs(1,1)
Hss(2,1,2)=Gs(1,2)
Hss(2,1,3)=Gs(1,3)
Hss(2,2,1)=Hss(2,1,2)
Hss(2,2,2)=Hss(2,1,1)
Hss(2,2,3)=ST1*CT2
Hss(2,3,1)=Hss(2,1,3)
Hss(2,3,2)=Hss(2,2,3)
Hss(3,2,2)=L3*CT2
Hss(3,2,3)=ST2
Hss(3,3,2)=Hss(3,2,3)
C   TRANSFORM THE H-FUNCTION FROM THE LEG FRAME TO THE
C   BASE FRAME
CALL ROT_MATRIX(N,R)
CALL GENERAL_DOT(3,3,R,3,Hss,Hphi)
RETURN
END

*****
*
*   SUBROUTINE CAL_G_Hphi_uu
*
*****

SUBROUTINE CAL_G_Hphi_uu
C   THIS SUBROUTINE TRANSFERS THE G- AND H-FUNCTIONS
C   TO REFERENCE THEM TO THE COMMON PLATFORM
C   COORDINATES
DIMENSION ZGu(3,6),ZGp_phi(3,3),ZHuu(3,6,6),
+          ZHp_phi(3,3,3),ZGphi_u(3,6),

```

```

+           ZHphi_uu(3,6,6),TGu(6,3),GpHuu(3,6,6),
+           TGu_DOT_Hphi(3,6,3),TGuHphiGu(3,6,6),
+           ZIGp_phi(3,3),ZGphi_p(3,3),ZHphi_pp(3,3,3)
COMMON/ARR_Gu/Gu(6,3,6)
COMMON/ARR_Huu/Huu(6,3,6,6)
COMMON/G_Hp_phi/Gp_phi(6,3,3),Hp_phi(6,3,3,3)
COMMON/ARR_phi_uu/Gphi_u(6,3,6),Hphi_uu(6,3,6,6)
COMMON/G_Hphi_pp/Gphi_p(6,3,3),Hphi_pp(6,3,3,3)
DO I=1,6
C   SET Gu AND Gp INTO 2-D ARRAY
C   SET Huu AND Hpp INTO 3-D ARRAY
      CALL CHG_DIMHI(I,0,Gu,Huu,ZGu,ZHuu)
      CALL CHG_DIMHI(I,3,Gphi_p,Hphi_pp,ZGphi_p,ZHphi_pp)
      CALL TRANSPOSE(3,6,ZGu,TGu)
      CALL MATRIX_MULT(3,3,ZGphi_p,3,6,ZGu,ZGphi_u)
      CALL GENERAL_DOT(3,3,ZGphi_p,6,ZHuu,GpHuu)
      DO J=1,3
        CALL NN_DOT_NNN(6,3,J,3,TGu,ZHphi_pp,
+           TGu_DOT_Hphi)
        CALL NNN_DOT_NN(6,3,J,3,TGu_DOT_Hphi,ZGu,
+           TGuHphiGu)
        CALL ADDT(3,6,J,GpHuu,TGuHphiGu,ZHphi_uu)
      ENDDO
C   SET Gphi_u INTO 3-D ARRAY AND Hphi_uu INTO 4-D
C   ARRAY
      CALL CHG_DIMLO(I,3,6,ZGphi_u,3,6,6,ZHphi_uu,
+           Gphi_u,Hphi_uu)
      ENDDO
      RETURN
      END

```

```

*****
*
*
*
*
*****
SUBROUTINE GET_CONST(Izz1,Izz2,Ixx2,Ixx3,Iyy3,
+ M23,M34,Lc,Lr)
C THIS SUBROUTINE CALCULATES THE INERTIA TENSORS FOR
C EACH LINK
C REFERENCED TO THEIR RESPECTIVE LOCAL COORDINATES
REAL Izz1,Izz2,Ixx2,Ixx3,Iyy3,M12,M23,M34,Lc,Lr
READ(40,*) M12,M23,M34,R,Lc,Lr
Izz1=(M12*R**2)/2.
Izz2=(M23*Lc**2)/12.
Ixx2=Izz2
Ixx3=(M34*Lr**2)/12.
Iyy3=Ixx3
RETURN
END
*****
*
*
*
*
*****
SUBROUTINE CAL_INERTIA
*****
SUBROUTINE CAL_INERTIA(N,Izz1,Izz2,Ixx2,Ixx3,Iyy3,
+ M23,M34,Lc,Lr,Iphi_LK)
C THIS SUBROUTINE CALCULATES THE EFFECTIVE INERTIA
C MATRIX FOR EACH LEG EXPRESSED IN THE RESPECTIVE
C LEG FRAME
REAL Izz1,Izz2,Ixx2,Ixx3,Iyy3,M23,M34,Lc,Lr,
+ Iphi_LK(6,3,3)
COMMON/TETA_L/ST2(6),CT2(6),RL3(6)

```

```

Iphi_LK(N,1,1)=Izz1+M23*(Lc*ST2(N)/2.)**2+
+           Ixx2*ST2(N)**2+M34*(RL3(N)-Lr/2.)**2*
+           ST2(N)**2+Ixx3*ST2(N)**2
Iphi_LK(N,2,2)=M23*(Lc/2.)**2+Izz2+M34*(RL3(N)-
+           Lr/2.)**2+Iyy3
Iphi_LK(N,3,3)=M34
RETURN
END
*****
*
*           SUBROUTINE CAL_POW_phi
*
*****

SUBROUTINE CAL_POW_phi(N,Ixx2,Ixx3,Iyy3,M23,M34,
+           Lc,Lr,Pphi_LK)
C   THIS SUBROUTINE CALCULATES THE INERTIA POWER
C   ARRAY FOR EACH LEG EXPRESSED IN THE RESPECTIVE
C   LEG FRAME
REAL Ixx2,Ixx3,Iyy3,M23,M34,Lc,Lr
DIMENSION Pphi_LK(6,3,3,3)
COMMON/TETA_L/S2(6),C2(6),RL3(6)
R=RL3(N)-Lr/2.
SC=S2(N)*C2(N)
Pphi_LK(N,1,1,2)=M23*SC*(Lc/2.)**2+2.*Ixx2*SC+
+           M34*R**2*SC+2.*Ixx3*SC
Pphi_LK(N,1,1,3)=M34*R*S2(N)**2
Pphi_LK(N,1,2,1)=M23*SC*(Lc/2.)**2+M34*SC*R**2
Pphi_LK(N,1,2,3)=-Iyy3*S2(N)
Pphi_LK(N,1,3,1)=M34*R*S2(N)**2
Pphi_LK(N,2,1,1)=-M23*SC*(Lc/2.)**2-Ixx2*SC-
+           M34*SC*R**2-Ixx3*SC
Pphi_LK(N,2,1,3)=Ixx3*S2(N)

```

```

Pphi_LK(N,2,2,3)=M34*R
Pphi_LK(N,2,3,2)=M34*R
Pphi_LK(N,3,1,1)=-M34*R*S2(N)**2
Pphi_LK(N,3,1,2)=-Ixx3*S2(N)
Pphi_LK(N,3,2,1)=Iyy3*S2(N)
Pphi_LK(N,3,2,2)=-M34*R
RETURN
END

```

```

*****

```

```

*
*
*           SUBROUTINE CAL_Iuu_Puuu
*
*

```

```

*****

```

```

      SUBROUTINE CAL_Iuu_Puuu
C      THIS SUBROUTINE CALCULATES THE TOTAL EFFECTIVE
C      INERTIA MATRIX AND INERTIA POWER ARRAY FOR THE
C      SIX LEGS REFERENCED TO THE COMMON PLATFORM
C      COORDINATES
      COMMON/ARR_phi_uu/Gphi_u(6,3,6),Hphi_uu(6,3,6,6)
      COMMON/SINERTIA_phi/SIphi_LK(6,3,3)
      COMMON/POWER_phi/Pphi_LK(6,3,3,3)
      COMMON/SINERTIA_u/SIuu(6,6,6)
      COMMON/POWER_u/Puuu(6,6,6,6)
      COMMON/ARR_Gu/Gu(6,3,6)
      COMMON/ARR_Huu/Huu(6,3,6,6)
      COMMON/G_Hphi_pp/Gphi_p(6,3,3),Hphi_pp(6,3,3,3)
      DIMENSION ZGphi_u(3,6),TGphi_u(6,3),ZHphi_uu(3,6,6),
+           TZGphi_u(6,3),ZSIphi_LK(3,3),
+           ZPphi_LK(3,3,3),ZHu_phi(6,3,3),ZSIuu(6,6),
+           ZPuuu(6,6,6),ZGphi_p(3,3),ZHphi_pp(3,3,3),
+           TZGphi_pp(3,3),W(3,3),WW(3,6,6),X(3,3,3),
+           TZGphi_p(3,3),ZGu(3,6),ZHuu(3,6,6),TGu(6,3),

```

```

+           Y(3,3,3),YY(3,3,3),X_YY(3,3,3),V(3,6,3),
+           VV(3,6,6),WV(3,6,6)
DO I=1,6
C   SET 3-D ARRAY INTO 2-D AND 4-D ARRAY INTO 3-D
    CALL CHG_DIMHI(I,0,Gphi_u,Hphi_uu,ZGphi_u,ZHphi_uu)
    CALL CHG_DIMHI(I,3,SIphi_LK,Pphi_LK,ZSIphi_LK,
+           ZPphi_LK)
    CALL CHG_DIMHI(I,0,Gu,Huu,ZGu,ZHuu)
    CALL CHG_DIMHI(I,3,Gphi_p,Hphi_pp,ZGphi_p,ZHphi_pp)
    CALL CAL_I(3,6,ZGphi_u,TZGphi_u,3,3,ZSIphi_LK,ZSIuu)
    CALL TRANSPOSE(3,3,ZGphi_p,TZGphi_p)
    CALL MATRIX_MULT(3,3,ZSIphi_LK,3,3,ZGphi_p,W)
    CALL GENERAL_DOT(3,3,W,6,ZHuu,WW)
    CALL GENERAL_DOT(3,3,ZSIphi_LK,3,ZHphi_pp,X)
    CALL TRANSPOSE(3,6,ZGu,TGu)
    DO J=1,3
        CALL NN_DOT_NNN(3,3,J,3,TZGphi_p,ZPphi_LK,Y)
        CALL NNN_DOT_NN(3,3,J,3,Y,ZGphi_p,YY)
    ENDDO
    DO J=1,3
        CALL ADDT(3,3,J,X,YY,X_YY)
        CALL NN_DOT_NNN(6,3,J,3,TGu,X_YY,V)
        CALL NNN_DOT_NN(6,3,J,3,V,ZGu,VV)
        CALL ADDT(3,6,J,WW,VV,WV)
    ENDDO
    CALL GENERAL_DOT(6,3,TZGphi_u,6,WV,ZPuuu)
    CALL CHG_DIMLO(I,6,6,ZSIuu,6,6,6,ZPuuu,SIuu,Puuu)
ENDDO
RETURN
END

```

```

*****
*
*                               SUBROUTINE CAL_I
*
*****

      SUBROUTINE CAL_I(I,J,A,TA,K,L,B,C)
C      THIS SUBROUTINE SERVES AS A GENERAL EXPRESSION TO C
CALCULATE THE EFFECTIVE INERTIA MATRIX
      DIMENSION A(I,J),TA(6,6),B(K,L),C(J,J),TAS(6,6)
      CALL TRANSPOSE(I,J,A,TA)
      CALL MATRIX_MULT(J,I,TA,K,L,B,TAS)
      CALL MATRIX_MULT(J,L,TAS,I,J,A,C)
      RETURN
      END
*****
*
*                               SUBROUTINE PFORM_INERTIA
*
*****

      SUBROUTINE PFORM_INERTIA(RR,PL_G,PF_Iuu)
C      THIS SUBROUTINE CALCULATES THE EFFECTIVE INERTIA
C      MATRIX OF THE PLATFORM
      COMMON/TRAN_P_G/R(3,3)
      REAL M
      DIMENSION PF_Iuu(6,6),PL_I(3,3),PL_G(3,3),
+          PP(3,3),TR(3,3)
      M=2250.
      PF_Iuu(1,1)=M
      PF_Iuu(2,2)=M
      PF_Iuu(3,3)=M
      PL_I(1,1)=(M*RR**2)/4.
      PL_I(2,2)=PL_I(1,1)

```

```

      PL_I(3,3)=(M*RR**2)/2.
C     TRANSFORM Ixx,Iyy AND Izz TO GLOBAL COOR
      CALL MATRIX_MULT(3,3,R,3,3,PL_I,PP)
      CALL TRANSPOSE(3,3,R,TR)
      CALL MATRIX_MULT(3,3,PP,3,3,TR,PL_G)
      DO K=1,3
        DO J=1,3
          PF_luu(K+3,K+3)=PL_G(K,J)
        ENDDO
      ENDDO
      RETURN
      END

```

```

*****

```

```

*
*
*           SUBROUTINE PFORM_INEPOW
*

```

```

*****

```

```

      SUBROUTINE PFORM_INEPOW(P_luu,P_uuu)
C     THIS SUBROUTINE CALCULATES THE INERTIA POWER
C     ARRAY OF THE PLATFORM
      DIMENSION P_luu(3,3),P_uuu(6,6,6)
      DO I=4,6
        J=I-3
        P_uuu(4,I,5)=P_luu(J,3)
        P_uuu(4,I,6)=-P_luu(J,2)
        P_uuu(5,I,4)=-P_luu(J,3)
        P_uuu(5,I,6)=P_luu(J,1)
        P_uuu(6,I,4)=P_luu(J,2)
        P_uuu(6,I,5)=-P_luu(J,1)
      ENDDO
      RETURN
      END

```

```

*****
*
*
*
*
*****

```

```

SUBROUTINE I_STAR_uu
C THIS SUBROUTINE CALCULATES THE EFFECTIVE INERTIA
C MATRIX OF THE GENERALIZED STEWART PLATFORM
C REFERENCED TO THE PLATFORM
COMMON/SINERTIA_u/SIuu(6,6,6)
DIMENSION STOT_Iuu(6,6),PF_Iuu(6,6),TOT_Iuu(6,6)
DO I=1,6
  DO J=1,6
    STOT_Iuu(I,J)=0.
    DO K=1,6
      STOT_Iuu(I,J)=SIuu(K,I,J)+STOT_Iuu(I,J)
    ENDDO
    TOT_Iuu(I,J)=STOT_Iuu(I,J)+PF_Iuu(I,J)
  ENDDO
ENDDO
RETURN
END

```

```

*****
*
*
*
*
*****

```

```

SUBROUTINE P_STAR_uuu(PF_Puuu,TOT_Puuu)
C THIS SUBROUTINE CALCULATES THE INERTIA POWER
C ARRAY OF THE GENERALIZED STEWART PLATFORM
C REFERENCED TO THE PLATFORM
COMMON/POWER_u/Puuu(6,6,6,6)

```

```

      DIMENSION PF_Puuu(6,6,6),STOT_Puuu(6,6,6),
+      TOT_Puuu(6,6,6)
      DO I=1,6
        DO J=1,6
          DO K=1,6
            STOT_Puuu(I,J,K)=0.
            DO L=1,6
              STOT_Puuu(I,J,K)=Puuu(L,I,J,K)+STOT_Puuu(I,J,K)
            ENDDO
            TOT_Puuu(I,J,K)=STOT_Puuu(I,J,K)+PF_Puuu(I,J,K)
          ENDDO
        ENDDO
      ENDDO
      RETURN
      END

```

```

*****

```

```

*
*
*      SUBROUTINE CAL_Gq_u
*
*

```

```

*****

```

```

      SUBROUTINE CAL_Gq_u
C      THIS SUBROUTINE DOES THE EXTRACTION OF THE
C      G-FUNCTION REFERENCED TO THE DESIRED JOINT SET OF
C      COORDINATES
      COMMON/ARR_phi_uu/Gphi_u(6,3,6),Hphi_uu(6,3,6,6)
      COMMON/ARR_Gq_u/Gq_u(6,6)
      WRITE(1,*)'Gq_u'
      DO I=1,6
        DO J=1,6
          Gq_u(I,J)=Gphi_u(I,3,J)
        ENDDO
      WRITE(1,99)(GQ_U(I,J),J=1,6)

```

```

      ENDDO
99  FORMAT(X,6(F10.3,2X))
      RETURN
      END

*****
*
*          SUBROUTINE CAL_Hq_uu
*
*****

      SUBROUTINE CAL_Hq_uu
C    THIS SUBROUTINE DOES THE EXTRACTION OF THE
C    H-FUNCTION REFERENCED TO THE DESIRED JOINT SET OF
C    COORDINATES
      COMMON/ARR_phi_uu/Gphi_u(6,3,6),Hphi_uu(6,3,6,6)
      COMMON/ARR_Hq_uu/Hq_uu(6,6,6)
      WRITE(1,*)'Hq_uu'
      DO I=1,6
        WRITE(1,*)'PLANE=',I
        DO J=1,6
          DO K=1,6
            Hq_uu(I,J,K)=Hphi_uu(I,3,J,K)
          ENDDO
          WRITE(1,99)(Hq_uu(I,J,K),K=1,6)
        ENDDO
      ENDDO
99  FORMAT(X,6(F10.3,2X))
      RETURN
      END

```

```

*****
*
*                               SUBROUTINE CAL_Iqq_Pqqq                               *
*
*****

      SUBROUTINE CAL_Iqq_Pqqq
C      THIS SUBROUTINE CALCULATES THE EFFECTIVE INERTIA
C      MATRIX AND INERTIA POWER ARRAY REFERENCED TO
C      THE DESIRED JOINT SET OF GENERALIZED COORDINATES
      COMMON/ARR_Gq_u/Gq_u(6,6)
      COMMON/ARR_Hq_uu/Hq_uu(6,6,6)
      COMMON/Iuu/TOT_Iuu(6,6)
      COMMON/Puuu/TOT_Puuu(6,6,6)
      COMMON/Iqq/STAR_Iqq(6,6)
      COMMON/Pqqq/P_STAR_qqq(6,6,6)
      DIMENSION Gu_q(6,6),TGu_q(6,6),EE(6,6,6),FF(6,6,6),
+              GG(6,6,6),XX(6,6,6),YY(6,6,6),Hu_qq(6,6,6),
+              F(6,6,6)
      CALL MATRIX_INV(6,Gq_u,Gu_q)
      CALL CAL_I(6,6,Gu_q,TGu_q,6,6,TOT_Iuu,STAR_Iqq)
      WRITE(1,*)'I*qq'
      DO II=1,6
        WRITE(1,*)(STAR_IQQ(II,JJ),JJ=1,6)
      ENDDO
      CALL CAL_Hu_qq(Gu_q,TGu_q,Hu_qq)
      CALL GENERAL_DOT(6,6,TGu_q,6,TOT_Puuu,EE)
      CALL GENERAL_DOT(6,6,STAR_Iqq,6,Hq_uu,FF)
      WRITE(1,*)'P*qqq'
      DO I=1,6
        WRITE(1,*)'PLANE=',I
        CALL SUBT(6,I,EE,FF,GG)
        CALL NN_DOT_NNN(6,6,I,6,TGu_q,GG,XX)

```

```

      CALL NNN_DOT_NN(6,6,I,6,XX,Gu_q,P_STAR_qqq)
      DO J=1,6
        WRITE(1,*)(P_STAR_QQQ(I,J,K),K=1,6)
      ENDDO
    ENDDO
  RETURN
END

```

```

*****

```

```

*
*
*          SUBROUTINE CAL_Hu_qq
*
*

```

```

*****

```

```

      SUBROUTINE CAL_Hu_qq(Gu_q,TGu_q,Hu_qq)
C    THIS SUBROUTINE CALCULATES THE DIRECT TRANSFER OF
C    Gq_u AND Hq_uu
      COMMON/ARR_Hq_uu/Hq_uu(6,6,6)
      DIMENSION Gu_q(6,6),TGu_q(6,6),GH(6,6,6),BB(6,6,6),
+          CC(6,6,6),Hu_qq(6,6,6)
      CALL GENERAL_DOT(6,6,Gu_q,6,Hq_uu,GH)
      DO I=1,6
        CALL NN_DOT_NNN(6,6,I,6,TGu_q,GH,BB)
        CALL NNN_DOT_NN(6,6,I,6,BB,Gu_q,CC)
        DO J=1,6
          DO K=1,6
            Hu_qq(I,J,K)=-CC(I,J,K)
          ENDDO
        ENDDO
      ENDDO
    RETURN
  END

```

```
*****
*
*                               SUBROUTINE CHG_DIMHI
*
*****
```

```
      SUBROUTINE CHG_DIMHI(M,N,A,B,C,D)
C      THIS SUBROUTINE CHANGES 3-DIMENSIONAL ARRAY TO
C      2-DIMENSIONAL MATRIX AND 4-DIMENSIONAL ARRAY TO
C      3-DIMENSIONAL ARRAY
      DIMENSION A(6,3,6-N),B(6,3,6-N,6-N),C(3,6-N),
+          D(3,6-N,6-N)
      DO J=1,3
        DO K=1,6-N
          C(J,K)=A(M,J,K)
          DO L=1,6-N
            D(J,K,L)=B(M,J,K,L)
          ENDDO
        ENDDO
      ENDDO
      RETURN
      END
```

```
*****
*
*                               SUBROUTINE CHG_DIMLO
*
*****
```

```
      SUBROUTINE CHG_DIMLO(II,JJ,KK,A,LL,MM,NN,B,C,D)
C      THIS SUBROUTINE CHANGES 2-DIMENSIONAL MATRIX TO
C      3-DIMENSIONAL ARRAY AND 3-DIMENSIONAL ARRAY TO
C      4-DIMENSIONAL ARRAY
      DIMENSION A(JJ,KK),B(LL,MM,NN),C(6,JJ,KK),D(6,LL,MM,NN)
      DO J=1,JJ
```

```

DO K=1, KK
  C(II,J,K)=A(J,K)
  DO L=1, NN
    D(II,J,K,L)=B(J,K,L)
  ENDDO
ENDDO
RETURN
END

```

```

*****

```

```

*
*
*          SUBROUTINE MATRIX_INV
*

```

```

*****

```

```

      SUBROUTINE MATRIX_INV(N,A,AI)
C     THIS SUBROUTINE FINDS THE INVERSE OF ANY SQUARE N
C     BY N MATRIX
      REAL A(N,N),AI(N,N)
      DIMENSION INTER(15,2)
C
      DO I=1,N
        DO J=1,N
          AI(J,I)=A(J,I)
        ENDDO
      ENDDO
C
      DO 12 K=1,N
        JJ=K
        IF (K.NE.N) THEN
          KP1=K+1
          BIG=ABS(AI(K,K))
          DO 5 I=KP1,N

```

```

        AB=ABS(AI(I,K))      ! SEARCHING FOR LARGEST PIVOT
        IF (BIG.LT.AB) THEN
            BIG=AB
            JJ=I
        ENDIF
5      CONTINUE
    ENDIF
C
    INTER(K,1)=K
    INTER(K,2)=JJ
    IF (JJ.NE.K) THEN
        DO 8 J=1,N
            TEMP=AI(K,J)
            AI(K,J)=AI(JJ,J)      ! INTERCHANGE ROWS
            AI(JJ,J)=TEMP
8      CONTINUE
    ENDIF
C
    DO 10 J=1,N
        IF (J.NE.K) THEN
            TEMP=AI(K,J)/AI(K,K)
            AI(K,J)=TEMP
        ENDIF
10     CONTINUE
    TEMP=1./AI(K,K)
    AI(K,K)=TEMP
    DO 11 I=1,N
        IF (I.NE.K) THEN
            DO J=1,N
                IF (J.NE.K) THEN
                    TEMP=AI(I,J)-AI(K,J)*AI(I,K)
                    AI(I,J)=TEMP

```

```

        ENDIF
        ENDDO
        ENDIF
11      CONTINUE
C
        DO 12 I=1,N
          IF (I.NE.K) THEN
            TEMP=-AI(I,K)*AI(K,K)
            AI(I,K)=TEMP
          ENDIF
12      CONTINUE
        DO 13 L=1,N
          K=N-L+1
          KROW=INTER(K,1)
          IROW=INTER(K,2)
          IF (KROW.NE.IROW) THEN
            DO I=1,N
              TEMP=AI(I,KROW)
              AI(I,KROW)=AI(I,IROW)
              AI(I,IROW)=TEMP
            ENDDO
          ENDIF
13      CONTINUE
        RETURN
        END

```

```

*****

```

```

*

```

```

*          SUBROUTINE GENERAL_DOT          *

```

```

*

```

```

*****

```

```

        SUBROUTINE GENERAL_DOT(N,NC,A,M,B,AB)

```

```

C      THIS SUBROUTINE PERFORMS THE GENERALIZED DOT

```

```

C      PRODUCT OF THE N BY NC MATRIX A AND NC BY M BY M
C      ARRAY B
      DIMENSION AB(N,M,M),A(N,NC),B(NC,M,M)
      DO I=1,N
        DO J=1,M
          DO K=1,M
            AB(I,J,K)=0
            DO L=1,NC
              AB(I,J,K)=A(I,L)*B(L,J,K)+AB(I,J,K)
            ENDDO
          ENDDO
        ENDDO
      ENDDO
      RETURN
      END

```

```

*****

```

```

*
*
*      SUBROUTINE TRANSPOSE
*
*

```

```

*****

```

```

      SUBROUTINE TRANSPOSE(M,N,A,AT)
C      THIS SUBROUTINE FINDS THE TRANSPOSE OF THE M BY N
C      MATRIX A
      DIMENSION A(M,N),AT(N,M)
      DO I=1,M
        DO J=1,N
          AT(J,I)=A(I,J)
        ENDDO
      ENDDO
      RETURN
      END

```

```
*****
*
*                               SUBROUTINE MATRIX_MULT
*
*****
```

```
      SUBROUTINE MATRIX_MULT(IA,J,T,K,L,F,TF)
C      THIS SUBROUTINE MULTIPLIES A IA BY J MATRIX WITH A
C      K BY L MATRIX F
      DIMENSION T(IA,J),F(K,L),TF(IA,L)
      INTEGER Q
      DO M=1,IA
        DO N=1,L
          TF(M,N)=0.0
          DO Q=1,K
            TF(M,N)=T(M,Q)*F(Q,N)+TF(M,N)
          ENDDO
        ENDDO
      ENDDO
      RETURN
      END
```

```
*****
*
*                               SUBROUTINE NNN_DOT_NN
*
*****
```

```
      SUBROUTINE NNN_DOT_NN(M,N,L,LL,A,B,A_DOT_B)
C      THIS SUBROUTINE MULTIPLIES A LL BY M BY N ARRAY A
C      WITH A N BY M MATRIX B
      DIMENSION A(LL,M,N),B(N,M),A_DOT_B(LL,M,M)
      DO I=1,M
        DO J=1,M
          A_DOT_B(L,I,J)=0.
        ENDDO
      ENDDO
```

```

      DO K=1,N
        A_DOT_B(L,I,J)=A(L,I,K)*B(K,J)+A_DOT_B(L,I,J)
      ENDDO
    ENDDO
  ENDDO
  RETURN
END

```

```

*****

```

```

*
*
*          SUBROUTINE NN_DOT_NNN
*
*

```

```

*****

```

```

      SUBROUTINE NN_DOT_NNN(M,N,L,LL,A,B,A_DOT_B)
C    THIS SUBROUTINE MULTIPLIES A M BY N MATRIX A WITH A
C    LL BY N BY N ARRAY B
      DIMENSION A(M,N),B(LL,N,N),A_DOT_B(LL,M,N)
      DO I=1,M
        DO J=1,N
          A_DOT_B(L,I,J)=0.
          DO K=1,N
            A_DOT_B(L,I,J)=A(I,K)*B(L,K,J)+A_DOT_B(L,I,J)
          ENDDO
        ENDDO
      ENDDO
      RETURN
END

```

```

*****

```

```

*
*
*          SUBROUTINE ADDT
*
*

```

```

*****

```

```

      SUBROUTINE ADDT(L,M,N,A,B,A_B)

```

```

C   THIS SUBROUTINE ADDS A L BY M BY M ARRAY A TO A
C   L BY M BY M ARRAY B
      DIMENSION A(L,M,M),B(L,M,M),A_B(L,M,M)
      DO I=1,M
        DO J=1,M
          A_B(N,I,J)=A(N,I,J)+B(N,I,J)
        ENDDO
      ENDDO
      RETURN
      END

```

```

*****

```

```

*
*
*           SUBROUTINE SUBT
*
*

```

```

*****

```

```

      SUBROUTINE SUBT(M,N,A,B,A_B)
C   THIS SUBROUTINE SUBTRACT A N BY M BY M ARRAY A
C   FROM A N BY M BY M ARRAY B
      DIMENSION A(6,6,6),B(6,6,6),A_B(6,6,6)
      DO I=1,M
        DO J=1,M
          A_B(N,I,J)=A(N,I,J)-B(N,I,J)
        ENDDO
      ENDDO
      RETURN
      END

```

```

*****

```

```

*
*
*           SUBROUTINE ROT_MATRIX
*
*

```

```

*****

```

```

      SUBROUTINE ROT_MATRIX(N,R)

```

```

C   THIS SUBROUTINE ROTATES THE LEG FRAMES TO THE BASE
C   FRAME
      DIMENSION R(3,3)
      DELTA=5.
      IF(N .EQ. 1) AL=0.
      IF(N .EQ. 2) AL=DELTA
      IF(N .EQ. 3) AL=120.
      IF(N .EQ. 4) AL=120.+DELTA
      IF(N .EQ. 5) AL=240.
      IF(N .EQ. 6) AL=240.+DELTA
      R(1,1)=1.
      R(2,2)=COSD(AL)
      R(2,3)=-SIND(AL)
      R(3,2)=-R(2,3)
      R(3,3)=R(2,2)
      RETURN
      END

```

```

*****
*                                                                 *
*                               SUBROUTINE TRANSF_P_G              *
*                                                                 *
*****

```

```

      SUBROUTINE TRANSF_P_G
C   THIS SUBROUTINE ROTATES THE PLATFORM FRAME TO THE
C   BASE FRAME
      COMMON/TRAN_P_G/R(3,3)
      COMMON/ORIEN/X1,Y1,Z1,X12,Y12,Z12
      R(1,1)=X12
      R(1,2)=Y1*Z12-Z1*Y12
      R(1,3)=X1
      R(2,1)=Y12
      R(2,2)=Z1*X12-X1*Z12

```

```
R(2,3)=Y1
R(3,1)=Z12
R(3,2)=X1*Y12-Y1*X12
R(3,3)=Z1
RETURN
END
```

```

*****
*
*
*          SUBROUTINE CAL_QDOT
*
*****

```

```

SUBROUTINE CAL_QDOT(UDOT,QDOT)
C THIS SUBROUTINE COMPUTES THE DESIRED ACTUATORS
C VELOCITY
COMMON/ARR_Gq_u/Gq_u(6,6)
DIMENSION UDOT(6,1),QDOT(6,1)
CALL MATRIX_MULT(6,6,Gq_u,6,1,UDOT,QDOT)
RETURN
END

```

```

*****
*
*
*          SUBROUTINE CAL_QDDOT
*
*****

```

```

SUBROUTINE CAL_QDDOT(UDOT,UDDOT,QDDOT)
C THIS SUBROUTINE COMPUTES THE DESIRED ACTUATORS
C ACCELERATION
COMMON/ARR_Gq_u/Gq_u(6,6)
COMMON/ARR_Hq_uu/Hq_uu(6,6,6)
DIMENSION UDOT(6,1),UDDOT(6,1),QDDOT(6,1),XX(6,1),
+          TUDOT(1,6),YY(6,1,6),ZZ(6,1,1)
CALL MATRIX_MULT(6,6,Gq_u,6,1,UDDOT,XX)

```

```

CALL TRANSPOSE(6,1,UDOT,TUDOT)
DO I=1,6
  CALL NN_DOT_NNN(1,6,I,6,TUDOT,Hq_uu,YY)
  CALL NNN_DOT_NN(1,6,I,6,YY,UDOT,ZZ)
  QDDOT(I,1)=XX(I,1)+ZZ(I,1,1)
ENDDO
RETURN
END

```

```

*****

```

```

*
*
*          SUBROUTINE GEN_FORCES
*
*
*****

```

```

      SUBROUTINE GEN_FORCES(QDOT,QDDOT)
C     THIS SUBROUTINE COMPUTES THE REQUIRED ACTUATOR
C     FORCES TO CAUSE THE DESIRED PLATFORM MOTIONS
      COMMON/Iqq/STAR_Iqq(6,6)
      COMMON/Pqqq/P_STAR_qqq(6,6,6)
      DIMENSION QDOT(6,1),TQDOT(1,6),QDDOT(6,1),XX(6,1),
+          TQ(6,1),YY(6,1,6),ZZ(6,1,1)
      CALL TRANSPOSE(6,1,QDOT,TQDOT)
      CALL MATRIX_MULT(6,6,STAR_Iqq,6,1,QDDOT,XX)
      DO I=1,6
        CALL NN_DOT_NNN(1,6,I,6,TQDOT,P_STAR_qqq,YY)
        CALL NNN_DOT_NN(1,6,I,6,YY,QDOT,ZZ)
        Tq(I,1)=ZZ(I,1,1)+XX(I,1)
      ENDDO
      WRITE(1,88)
      DO I=1,6
        WRITE(1,99)I,XX(I,1),ZZ(I,1,1),TQ(I,1)
      ENDDO
88    FORMAT(/,X,'LINK',10X,'ACC. TERMS',10X,'VEL.

```

```
      +          TERMS',10X,'Tq')  
99    FORMAT(3X,I1,8X,E15.8,5X,E15.8,4X,E15.8)  
      RETURN  
      END
```

**APPENDIX C**  
**INPUT DATA FOR THE SIMULATION**  
**PROGRAM IN APPENDIX B**

1, 0, 0, 0, 1, 0, 0

3, 0, 0

1, 0, 0, 0, 0.5, -0.866, 60

4, 0, 0

2, 3

10, 200, 175, 0.06, 3, 3

2

20

## APPENDIX D

### SUBROUTINE FOR CLASS P=2, CONSTANT ACCELERATION

```

*****
*
*                               SUBROUTINE NEXT_MOTION
*
*****
SUBROUTINE NEXT_MOTION(TMAX,PTIME,XUi,XUf,THETAi,
+                               THETAf,XU,THETA,V,W,ACC,
+                               AL,Y12,Z12)
C THIS SUBROUTINE DETERMINES THE PLATFORM MOTIONS
C USING CLASS P=2, CONSTANT ACCELERATION
  AUG=(PTIME-TMAX/2)
  T=TMAX**2
  IF (AUG .LT. 0) THEN
    AUG=0.
    DEL=0.
  ELSE
    DEL=1.
  ENDIF
  X=XUf-XUi
  TTA=THETAf-THETAi
  XU=XUi+2.*X*(PTIME**2-2.*AUG**2)/T
  V=4.*X*(PTIME-2.*AUG)/T
  ACC=4.*X*(1.-2.*DEL)/T
  THETA=THETAi+2.*TTA*(PTIME**2-2.*AUG**2)/T
  W=4.*TTA*(PTIME-2.*AUG)/T
  AL=4.*TTA*(1.-2.*DEL)/T
  Y12=COS(THETA)
  Z12=SIN(THETA)
  THETA=THETA*180./3.1415926
  RETURN
  END

```

**SUBROUTINE FOR CLASS P=4,  
4-5-6-7 POLYNOMIAL**

202

APPENDIX F  
SUBROUTINE FOR CLASS P=4,  
THIRD DERIVATIVE TRAPEZOIDAL

```

*****
*
*                               SUBROUTINE NEXT_MOTION
*
*****
SUBROUTINE NEXT_MOTION(TMAX,PTIME,XUi,XUf,THETAi,
+                               THETAf,XU,THETA,V,W,ACC,AL,
+                               Y12,Z12,T,A1,A2,A3)
C   THIS SUBROUTINE DETERMINES THE PLATFORM MOTIONS
C   USING CLASS P=4, THIRD DERIVATIVE TRAPEZOIDAL
DIMENSION T(0:8),AUG(0:7),Y(2,4),A1(2),A2(2),A3(2)
DO II=1,2
  DO I=4,2,-1
    DO J=1,8
      AUG(J-1)=PTIME-T(J-1)
      IF (AUG(J-1) .LT. 0.)AUG(J-1)=0.
      AUG(J-1)=AUG(J-1)**I
    ENDDO
    FACT=I
    DO K=I-1,1,-1
      FACT=FACT*K
    ENDDO
    Y(II,I)=(A1(II)/FACT)*((AUG(0)-AUG(1))/(T(1)-T(0))-
+      (AUG(2)-AUG(3))/(T(3)-T(2)))+(A2(II)/FACT)*
+      ((AUG(2)-AUG(3))/(T(3)-T(2))-(AUG(4)-AUG(5))/
+      (T(5)-T(4)))+(A3(II)/FACT)*((AUG(4)-AUG(5))
+      /(T(5)-T(4))-(AUG(6)-AUG(7))/(TMAX-T(6)))
    ENDDO
  ENDDO
  XU=XUi+Y(1,4)
  V=Y(1,3)
  ACC=Y(1,2)
  THETA=THETAi+Y(2,4)
  W=Y(2,3)
  AL=Y(2,2)
  Y12=COS(THETA)
  Z12=SIN(THETA)
  THETA=THETA*180./3.1415926
  RETURN
END

```

## APPENDIX G

### SUBROUTINE TO DETERMINE THE CONSTANTS A1, A2 AND A3

```

*****
*
*                               SUBROUTINE CAL_A1A2A3
*
*****
      SUBROUTINE CAL_A1A2A3(T,A1,A2,A3,XUf,THETAf,
+                               THETAi)
C   THIS SUBROUTINE DETERMINES THE THREE CONSTANTS A1,A2
C   AND A3 FOR THE CLASS P=4, THIRD DERIVATIVE
C   TRAPEZOIDAL
      DIMENSION T(0:8),A1(2),A2(2),A3(2),TT(3,3),TTI(3,3),Z(3),A(3)
      DO I=1,3
        DO J=1,3
          TT(I,J)=((((T(7)-T(2*J-2))**(I+1))-(T(7)-T(2*J-1))**(I+1))
+                /(T(2*J-1)-T(2*J-2)))-((((T(7)-T(2*J))**(I+1))-
+                (T(7)-T(2*J+1))**(I+1))/(T(2*J+1)-T(2*J)))
        ENDDO
      ENDDO
      DO K=1,2
        IF (K.EQ. 1) THEN
          Y=XUf-XUi
        ELSE
          Y=THETAf-THETAi
        ENDIF
        Z(3)=24.*Y
        CALL MATRIX_INV(3,TT,TTI)
        CALL MATRIX_MULT(3,3,TTI,3,1,Z,A)
        A1(K)=A(1)
        A2(K)=A(2)
        A3(K)=A(3)
        WRITE(1,*)'A1=',A1(K),'A2=',A2(K),'A3=',A3(K)
      ENDDO
      RETURN
      END

```

## REFERENCES

- [1] Benedict, C. E., and Tesar, D., "Dynamic Response Analysis of Quasi-Rigid Mechanical Systems Using Kinematic Influence Coefficients," ASME Journal of Mechanisms, Vol. 6, 1971, p. 383-403.
- [2] Benedict, C. E., and Tesar, D., "Model Formulation of Complex Mechanisms with Multiple Inputs: Part I -- Geometry," ASME Journal of Mechanical Design, Vol. 100, October 1978a, p. 747-753.
- [3] Benedict, C. E., and Tesar, D., "Model Formulation of Complex Mechanisms with Multiple Inputs: Part II -- The Dynamic Model," ASME Journal of Mechanical Design, Vol. 100, October 1978b, p. 755-761.
- [4] Callan, J. F., "The Simulation and Programming of Multiple-Arm Robot System," Robotics Engineering, Vol. 8, No. 4, April 1986.
- [5] Colson, J. C., and Ferreira, N. D., "Kinematic Arrangements Used in Industrial Robots," 13<sup>th</sup> Industrial Robots Conference Proceedings, April 1983, p 20-1 - 20-8.
- [6] Cox, D. J., "The Dynamic Modeling and Command Signal Formulation for Parallel Multi-Parameter Robotic Devices," Master's Thesis, University of Florida, Gainesville, September 1981.
- [7] Craig, J. J., "Introduction to Robotics," Addison-Wesley Publishing Company, Reading, Massachusetts, 1986.
- [8] Cwiakala, M., "Workspace of a Closed-Loop Manipulator," ASME Journal of Mechanical Design, October 1986.
- [9] Do, W. Q. D., and Yang, D. C. H., "Inverse Dynamics of a Platform Type of Manipulator Structure," ASME Journal of Mechanical Design, October 1986.

- [10] Driga, M. D., Eppes, D., and Flake, R. H., "Electromechanical Energy Conversion Devices as Joint Actuators in Robotics and Automated Manufacturing," IEEE Region 5 Conference, March 1987.
- [11] Duffy, J., "Analysis of Mechanisms and Robot Manipulators," Edward Arnold (Publishers) Ltd., Bedford Square, London, 1980.
- [12] Fichter, E. F., "A Stewart Platform-Based Manipulator General Theory and Practical Construction," The International Journal of Robotics Research, Vol. 5, No. 2, Summer 1986.
- [13] Freeman, R. A., and Tesar, D., "The Generalized Coordinate Selection for the Dynamics of Complex Planar Mechanical Systems," ASME Journal of Mechanical Design, Vol. 104, January 1982, p. 206-217.
- [14] Freeman, R. A., and Tesar, D., "Kinematic and Dynamic Modeling, Analysis and Control of Robotic Mechanisms (via Generalized Coordinate Transformation)," Manufacturing Systems Engineering, Department of Mechanical Engineering, The University of Texas at Austin, June 1985. Based on a dissertation presented to the University of Florida, April 1985.
- [15] Freeman, R.A., and Tesar, D., "Dynamic Modeling for the Overconstrained Mode of Coordinated, Multiple-Manipulator Operation," Transactions of the 10<sup>th</sup> U.S. National Congress of Applied Congress of Applied Mechanics, Austin, TX, June 16-20, 1986.
- [16] Freeman, R. A., "Independence of Transfer Process on Non-actuated Ball Joint Connectors," Internal Document, Mechanical Engineering Department, The University of Texas at Austin, January 1987.
- [17] Fu, K. S., Gonzalez, R. C., and Lee, C. S. G., Robotics, McGraw-Hill Book Company, New York, New York, 1987.

- [18] Gates, R. M., and Graves, D. L., "Mathematical Model for the Simulation of Dynamic Docking Test System(DDTS) Active Table Motion," Boeing Aerospace Company, Document No. D2-118544-1, August 1974.
- [19] Greenwood, D. T., Principles of Dynamics, Prentice-Hall, Inc., Englewood Cliffs, New Jersey, 1965.
- [20] Hollerbach, J. M., "A Recursive Lagrangian Formulation of Manipulator Dynamics and a Comparative Study of Dynamics Formulation Complexity," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-10, No. 11, November 1980.
- [21] Hudgens, J. C., "Modeling and Analysis of a Fully-Parallel Six Degree-of-Freedom Micromanipulator," Master's Thesis, University of Florida, December 1986.
- [22] Hunt, K. H., "Structural Kinematics of In-Parallel-Actuated Robot-Arms," ASME Journal of Mechanisms, Transmissions, and Automation in Design, Vol. 105, December 1983, p.705-712.
- [23] Kane, T. R., and Levinson D. A., "The Use of Kane's Dynamical Equations in Robotics," The International Journal of Robotics Research, Vol. 2, No. 3, Fall 1983.
- [24] Lee, C. S. G., "Robot Arm Kinematics, Dynamics, and Control," IEEE Computer, Vol. 15, No. 12, December 1982, p. 62-80.
- [25] Lee, C. S. G., Lee, B. H., and Nigam, R., "Development of the Generalized d'Alembert Equations of Motion for Mechanical Manipulators," *Proceedings of the 22<sup>nd</sup> Conference on Decision and Control*, December 14-16, 1983.

- [26] Lee, C. S. G., and Ziegler, M., "Geometric Approach in Solving Inverse Kinematics of PUMA Robots," IEEE Transactions on Aerospace and Electronic Systems, Vol. AES-20, No. 6, November 1984, p. 695-706.
- [27] Luh, J. Y. S., Walker, M. W., and Paul, R. P. C., "Resolved-Acceleration Control of Mechanical Manipulators," IEEE Transactions on Automatic Control, Vol. AC-25, No. 3, June 1980, p. 468-474.
- [28] Marco, D., "Computer Simulation and Design of a Three Degree of Freedom Robotic Shoulder Module," Master's Thesis, The University of Texas at Austin, August, 1987.
- [29] Matthew, G. K., and Tesar, D., The Dynamic Synthesis, Analysis, and Design of Modeled Cam Systems, Lexington Books, D. C. Heath and Company, Lexington, Massachusetts, 1976.
- [30] Mayer, G., and Wood, E. I., "Multiple-Arm Control and Assembly Operation," Robotics Engineering, Vol. 8, No. 4, April 1986.
- [31] Mohamed, M. G., and Duffy, J., "A Direct Determination of the Instantaneous Kinematics of Fully Parallel Robot Manipulators," ASME Design Engineering Technical Conference, ASME Paper 84-DET-114, Cambridge, MA, October 7-10, 1984.
- [32] Ogata, K., Modern Control Engineering, Prentice-Hall of India, New Delhi, 1986.
- [33] Owen, B., and Williams, E., "Dynamic Docking Test System Summary Document," Boeing Aerospace Company, Document No. D2-118568-1, August 1975.
- [34] Paul, R. P., Robot Manipulators, The MIT Press, Cambridge, Massachusetts, 1981.

- [35] Paul, R. P., Shimano, B., and Mayer, G., "Kinematic Control for Simple Manipulators," *IEEE Transactions on Systems, Man, and Cybernetics*, Vol. SMC-11, No. 6, June 1981.
- [36] Sklar, M. S., and Tesar, D., "Dynamic Analysis of Hybrid Serial Manipulator Systems Containing Parallel Modules," ASME Journal of Mechanisms, Transmissions, and Automation in Design, 1986.
- [37] Stewart, D., "A Platform with Six Degrees of Freedom," Proc. Institution of Mechanical Engineers, Vol. 180, Part 1, No. 15, 1965-66, p. 371-386.
- [38] Strassner, B. H., "Dynamic Docking Test System Final Mathematical Model," NASA Lyndon B. Johnson Space Center, Control Systems Development Division Internal Note 74-EG-23, July 1974.
- [39] Strassner, B. H., "Dynamic Docking Test System Initial Condition Model," NASA Lyndon B. Johnson Space Center, Control Systems Development Division Internal Note 74-EG-24, July 1974.
- [40] Strassner, B. H., "Dynamic Docking Test System Simulation Description," NASA Lyndon B. Johnson Space Center, Control Systems Development Division Internal Note 74-EG-25, August 1974.
- [41] Takano, M., Yashima, K., and Yada, S., "Development of Computer Simulation System of Kinematics and Dynamics of Robot," Journal of the Faculty of Engineering, The University of Tokyo, Vol. XXXVI, No. 4, 1982.
- [42] Thomas, M., and Tesar, D., "Dynamic Modeling and Analysis of Rigid-Link Serial Manipulators," DOE Grant DE-AC05-79ER10013, Center for Intelligent Machines and Robotics, University of Florida, Gainesville, Florida, November, 1982.

- [43] Thomas, M., and Tesar, D., "Dynamic Modeling of Serial Manipulator Arms," ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 104, September 1982, p. 218-228.
- [44] Torby, B. J., Advanced Dynamics for Engineers, Holt, Rinehart and Winston, New York, New York, 1984.
- [45] Tourassis, V. D., and Neuman, C. P., "Properties and Structure of Dynamic Robot Models for Control Engineering Applications," Mechanism and Machine Theory, Vol. 20, No. 1, p. 27-40, 1985.
- [46] Tourassis, V. D., and Neuman, C. P., "The Inertial Characteristics of Dynamic Robot Models," Mechanism and Machine Theory, Vol. 20, No. 1, p. 41-52, 1985.
- [47] Walker, M. W., and Orin, D. E., "Efficient Dynamic Computer Simulation of Robotic Mechanisms," ASME Journal of Dynamic Systems, Measurement, and Control, Vol. 104, September 1982, p. 205-211.
- [48] Whitney, D. E., "The Mathematics of Coordinated Control of Prosthetic Arms and Manipulators," ASME Journal of Dynamic Systems, Measurement, and Control, p. 303-309, December 1972.